

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO
FAKULTETA ZA MATEMATIKO IN FIZIKO

Tadej Borovšak

Intervalske Bézierove krivulje in ploskve

DIPLOMSKO DELO
UNIVERZITETNI ŠTUDIJSKI PROGRAM PRVE
STOPNJE RAČUNALNIŠTVO IN MATEMATIKA

MENTOR: doc. dr. Marjetka Krajnc

Ljubljana 2014

To delo je ponujeno pod licenco *Creative Commons Priznanje avtorstva-Deljenje pod enakimi pogoji 2.5 Slovenija* (ali novejšo različico). To pomeni, da se tako besedilo, slike, grafi in druge sestavine dela kot tudi rezultati diplomskega dela lahko prosto distribuirajo, reproducirajo, uporabljajo, priobčujejo javnosti in predelujejo, pod pogojem, da se jasno in vidno navede avtorja in naslov tega dela in da se v primeru spremembe, preoblikovanja ali uporabe tega dela v svojem delu, lahko distribuira predelava le pod licenco, ki je enaka tej. Podrobnosti licence so dostopne na spletni strani creativecommons.si ali na Inštitutu za intelektualno lastnino, Streliška 1, 1000 Ljubljana.



Izvorna koda diplomskega dela, njeni rezultati in v ta namen razvita programska oprema je ponujena pod licenco GNU General Public License, različica 3 (ali novejša). To pomeni, da se lahko prosto distribuira in/ali predeluje pod njenimi pogoji. Podrobnosti licence so dostopne na spletni strani <http://www.gnu.org/licenses/>.

Besedilo je oblikovano z urejevalnikom besedil \LaTeX .

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

Intervalska aritmetika nam omogoča enostavno spremljanje in kontrolo napake v numeričnih algoritmi. Intervalski polinom je polinom, katerega koeficienti so intervali. Ker so pri aproksimaciji krivulj v računalniško podprtem geometrijskem oblikovanju Bézierove krivulje eno izmed glavnih orodij, nas zanima posplošitev le teh na intervalsko aritmetiko. V diplomskem delu predstavite intervalske Bézierove krivulje in ploskve. Opišite posplošitve najbolj znanih algoritmov kot so de Casteljaujev algoritem, subdivizija in višanje stopnje na intervalsko aritmetiko ter jih ponazorite z zgledi. Postopke tudi implementirajte v programskem paketu Octave.

IZJAVA O AVTORSTVU DIPLOMSKEGA DELA

Spodaj podpisani Tadej Borovšak, z vpisno številko **63110426**, sem avtor diplomskega dela z naslovom:

Intervalske Bézierove krivulje in ploskve

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom doc. dr. Marjetke Krajnc,
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela,
- soglašam z javno objavo elektronske oblike diplomskega dela na svetovnem spletu preko univerzitetnega spletnega arhiva.

V Ljubljani, dne 17. septembra 2014

Podpis avtorja:

Kazalo

Povzetek

Abstract

| | | |
|----------|---|-----------|
| 1 | Uvod | 1 |
| 2 | Osnove | 3 |
| 2.1 | Bernsteinovi bazni polinomi | 3 |
| 2.2 | Intervalska aritmetika | 5 |
| 2.3 | Vektorski intervali | 6 |
| 2.4 | Intervalski polinomi | 8 |
| 3 | Intervalske Bézierove krivulje | 11 |
| 3.1 | Definicija intervalske Bézierove krivulje | 11 |
| 3.2 | De Casteljauev algoritem | 14 |
| 3.3 | Višanje stopnje Bézierove krivulje | 18 |
| 3.4 | Odvod Bézierove krivulje | 22 |
| 3.5 | Izračun večih točk na Bézierovi krivulji | 23 |
| 3.6 | Izris intervalskih Bézierovih krivulj | 26 |
| 4 | Intervalske Bézierove ploskve | 33 |
| 4.1 | Ploskve iz tenzorskega produkta | 33 |
| 4.2 | Intervalske Bézierove ploskve | 34 |
| 4.3 | Subdivizija | 35 |
| 4.4 | Višanje stopnje | 37 |

| | | |
|----------|---|-----------|
| 4.5 | Izris | 37 |
| 5 | Implementacija algoritmov in rezultati | 39 |
| 5.1 | Izračun točk | 39 |
| 5.2 | Prikaz rezultatov | 40 |
| 5.3 | Ravninske krivulje | 40 |
| 5.4 | Krivulje v prostoru | 44 |
| 5.5 | Ploskve | 47 |
| 6 | Zaključek | 49 |

Seznam uporabljenih kratic

| | |
|-------------|---|
| CAD | oblikovanje s pomočjo računalnika (<i>computer-aided design</i>) |
| CAM | z računalnikom podprta proizvodnja (<i>computer-aided manufacturing</i>) |
| CAGD | geometrijsko oblikovanje s pomočjo računalnika (<i>computer-aided geometric design</i>) |

Povzetek

Med oblikovanjem in izdelavo predmetov s pomočjo računalnika moramo podatke večkrat prenašati med različnimi CAD/CAM sistemi. Zaradi razlik med zmogljivostmi posameznih sistemov moramo podatke med prenosom aproksimirati. V diplomskem delu si bomo ogledali intervalske Bézierove krivulje in ploskve, ki predstavljajo eno izmed mnogih aproksimacijskih shem, razvitih v ta namen. Na kratko si bomo ogledali intervalsko aritmetiko ter opisali osnovne lastnosti intervalskih Bézierovih krivulj. Ogledali so bomo osnovne algoritme, ki jih uporabljamo pri delu z Bézierovimi krivuljami in ploskvami, in jih posplošili na intervalske Bézierove krivulje ter ploskve. Vse algoritme bomo implementirali v programskem jeziku Octave oziroma Matlab in grafično prikazali rezultate, ki jih dobimo z uporabo algoritmov na izbranih primerih.

Ključne besede: Bézierove krivulje, Bézierove ploskve, intervalska aritmetika.

Abstract

Process of creating real world objects usually involves usage of more than one CAD/CAM system. Communicating product data between diverse systems that use incompatible internal representation schemes necessitates data conversion and approximation. Main focus of the thesis will be on Bézier curves and surface patches that offer a way of converting data while retaining information about the approximation error made during conversion. We will briefly review basic building blocks such as intervals, interval arithmetic and Bernstein basis functions, and then resume with describing interval Bézier curves and surface patches. Crucial algorithms that allow us to work with Bézier curves, Bézier surface patches and their interval counterparts will be implemented. We will conclude this thesis with short review of a few selected examples of interval Bézier curves and surface patches.

Keywords: Bézier curves, Bézier surface patches, interval arithmetic.

Poglavje 1

Uvod

Skoraj vsak predmet iz našega vsakdanjega življenja vsaj del svojega obstoja dolguje oblikovalcem, ki so si ga zamislili, in strojem, ki so pomagali pri njegovi izdelavi. Zakaj so se stvari razvile ravno v to smer? Gotovo je eno izmed glavnih gonil človekova želja doseči čim boljše rezultate s čim manj opravljenega dela.¹ Zato tudi ni presenetljivo, da se je oblikovanje in izdelava predmetov s pomočjo računalnika začelo razvijati v avtomobilski industriji, kjer je bila osnovna naloga izdelati velike količine enakih avtomobilov.

Osnovna težava, ki jo skuša rešiti področje geometrijskega oblikovanja s pomočjo računalnika, je poiskati načine za predstavitev geometrijskih oblik z relativno majhno količino podatkov. Francoska matematika Pierre Étienne Bézier pri Renaultu in Paul de Casteljau pri Citroënu sta za ta namen (neodvisno) razvila tip parametričnih polinomskih krivulj in ploskev, ki danes nosijo ime Bézierove krivulje oziroma ploskve.

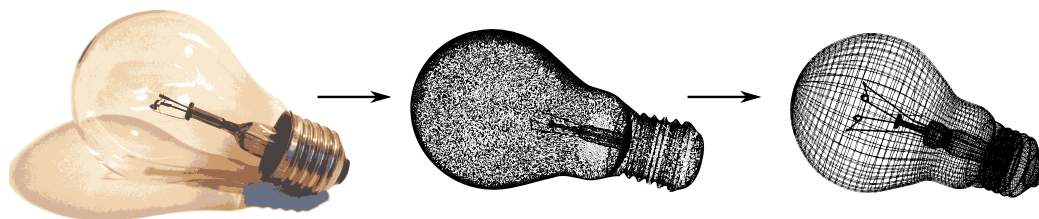
Uporaba Bézierovih krivulj se je iz avtomobilske industrije razširila takorekoč na vsa področja oblikovanja. V programih za obdelavo slik (npr. Inkscape ali Adobe Illustrator) je kubični Bézierov zlepek osnovna enota risbe, Bézierova konstrukcija krivulje s pomočjo ročic pa predstavlja način za spreminjanje oblike krivulje. Pri animaciji se te krivulje uporabljajo za določanje poti gi-

¹Iskanje odgovora na vprašanje, ali je to znamenje lenobe ali intelligence, bomo prepustili kakšni drugi znanstveni vedi.

banja, vsaka črka v tem besedilu pa je svoj obstoj pričela kot kvadratični Bézierov zlepek.

Pri oblikovanju in izdelavi predmetov s pomočjo računalnika nam le redko zadostuje en sam sistem. Komunikacija med sistemi z različnimi lastnostmi in zmogljivostmi pa med drugim pomeni tudi pretvorbe informacije. Vzemimo kot primer izdelavo modela žarnice na sliki 1.1. 3D skener najprej zajame podatke v obliki množice točk. Množico točk se pretvori v Bézierove ploskve, ki nato služijo kot navodilo stroju za izdelavo kalupa.

Za pretvorbo geometrijske informacije iz ene oblike v drugo obstaja mnogo načinov. Večina pretvorb nam zagotavlja, da bo dobljeni rezultat znotraj dogovorjenega tolerančnega območja, podatki o aproksimacijski napaki pa se med pretvorbo izgubijo in niso na voljo v nadaljnjih korakih. Intervalske Bézierove krivulje, katerih lastnosti si bomo ogledali v poglavju 3, pa informacijo o aproksimacijski napaki vsebujejo v svojem zapisu, kar omogoča tolerančno analizo tudi v kasnejših korakih.



Slika 1.1: Potek izdelave digitalnega modela žarnice.

Poglavje 2

Osnove

V tem poglavju si bomo na kratko osvežili znanje o Bernsteinovih baznih polinomih, si ogledali osnovne aritmetične operacije nad skalarnimi in vektorskimi intervali ter intervalske polinome.

2.1 Bernsteinovi bazni polinomi

Bernsteinovi bazni polinomi imajo osrednjo vlogo pri definiciji Bézierovih krivulj. Ker bomo v nadaljevanju potrebovali nekatere njihove posebne lastnosti, si pogledjmo njihovo definicijo z malo drugačnega zornega kota kot ponavadi.

Naj bo $t \in \mathbb{R}$ in $n \in \mathbb{N}_0$. Takrat velja

$$1 = 1^n = (1 - t + t)^n = (t + (1 - t))^n. \quad (2.1)$$

Z uporabo binomskega izreka lahko enačbo (2.1) zapišemo kot

$$(t + (1 - t))^n = \sum_{i=0}^n \binom{n}{i} t^i (1 - t)^{n-i}, \quad (2.2)$$

i -ti Bernsteinov bazni polinom stopnje n pa je definiran kot i -ti člen vsote v enačbi (2.2) in ga označimo z

$$B_i^n(t) = \binom{n}{i} t^i (1 - t)^{n-i}.$$

Zgornja konstrukcija nam zelo nazorno pokaže, da Bernsteinovi bazni polinomi tvorijo particijo enote ($\sum_{i=0}^n B_i^n(t) = 1$). Dodatno lahko še opazimo, da za $t \in [0, 1]$ velja tudi $B_i^n(t) \geq 0$, kar nam bo poenostavilo nekatere izračune v nadaljevanju.

Izpeljimo še dve enakosti, ki nam povesta, kako lahko Bernsteinov bazni polinom stopnje n izrazimo z Bernsteinovim baznim polinomom stopnje $n+1$. Z upoštevanjem definicije binomskega simbola lahko $B_i^n(t)$ zapišemo kot

$$\begin{aligned} B_i^n(t) &= \binom{n}{i} t^i (1-t)^{n-i} \\ &= \frac{n(1-t)}{n-i} \binom{n-1}{i} t^i (1-t)^{n-1-i} \\ &= \frac{n(1-t)}{n-i} B_i^{n-1}(t) \end{aligned} \quad (2.3)$$

ali

$$\begin{aligned} B_i^n(t) &= \binom{n}{i} t^i (1-t)^{n-i} \\ &= \frac{nt}{i} \binom{n-1}{i-1} t^{i-1} (1-t)^{n-i} \\ &= \frac{nt}{i} B_{i-1}^{n-1}(t). \end{aligned} \quad (2.4)$$

Še ena stvar, ki jo bomo potrebovali kasneje, je odvod Bernsteinovega baznega polinoma. Velja

$$\begin{aligned} \frac{d}{dt} B_i^n(t) &= \frac{d}{dt} \binom{n}{i} t^i (1-t)^{n-i} \\ &= \binom{n}{i} i t^{i-1} (1-t)^{n-i} - \binom{n}{i} (n-i) t^i (1-t)^{n-i-1} \\ &= \frac{ni}{i} \binom{n-1}{i-1} t^{i-1} (1-t)^{n-i} - \frac{(n-i)n}{n-i} \binom{n-i}{i} t^i (1-t)^{n-i-1} \\ &= n B_{i-1}^{n-1}(t) - n B_i^{n-1}(t). \end{aligned} \quad (2.5)$$

Višje odvode lahko izračunamo z rekurzivno uporabo zgornje formule.

2.2 Intervalska aritmetika

Intervalska aritmetika je orodje, s katerim lahko spremljamo propagacijo napak v izračunih. Napovedi napak, ki jih dobimo z uporabo intervalske aritmetike, so vedno točne, a pogosto pesimistične (napake v zaporedju izračunov se večkrat med seboj delno izničijo, intervalska aritmetika pa obravnava razplet po najslabšem mogočem scenariju). Na tem mestu si bomo ogledali le del, ki ga bomo potrebovali v nadaljevanju, podroben pregled področja pa lahko najdemo v [5] in [6].

Naj bo $[a, b] \subset \mathbb{R}$ zaprti interval, definiran kot

$$[a, b] = \{ t \mid a \leq t \leq b \}.$$

Zaradi poenostavitve nekaterih izračunov bomo skalarje pisali kot izrojen interval, $\forall a \in \mathbb{R} : a = [a, a]$. Nad dvema takšnima intervaloma $[a, b]$ in $[c, d]$ lahko sedaj definiramo operacijo $*$ $\in \{+, -, \cdot, /\}$ kot

$$[a, b] * [c, d] = \{ x * y \mid x \in [a, b], y \in [c, d] \}.$$

Z nekaj računanja lahko za zgoraj naštetih štiri osnovne operacije izpeljemo specifična pravila, ki nam omogočajo dejansko računanje z intervali:

$$\begin{aligned} [a, b] + [c, d] &= [a + c, b + d], \\ [a, b] - [c, d] &= [a - d, b - c], \\ [a, b] \cdot [c, d] &= [\min(ac, ad, bc, bd), \max(ac, ad, bc, bd)], \\ [a, b]/[c, d] &= [a, b] \cdot [1/c, 1/d], \end{aligned}$$

kjer deljenje definiramo le za intervale $[c, d]$, za katere velja $0 \notin [c, d]$.

Iz lastnosti osnovnih operacij lahko opazimo, da sta operaciji seštevanja in množenja komutativni in asociativni, vendar pa v splošnem nista distributivni. Poseben primer je množenje s skalarjem oziroma z izrojenim intervalom, ki pa je distributivno: $a([c, d] + [e, f]) = a[c, d] + a[e, f]$. Za poenostavitev izračunov uvedimo še krajšo oznako intervalov $[u]$, ki pomeni $[u] = [a, b]$.

Končno si oglejmo še nekaj izračunov oziroma enakosti, ki jih bomo potrebovali kasneje. Naj velja $a, b, c, d, e, f \in \mathbb{R}$, $a \geq 0$ in $b < 0$. Potem veljata naslednji enakosti:

$$a[c, d] = [a, a] \cdot [c, d] = [\min(ac, ad), \max(ac, ad)] = [ac, ad], \quad (2.6)$$

$$b[c, d] = [b, b] \cdot [c, d] = [\min(bc, bd), \max(bc, bd)] = [bd, bc]. \quad (2.7)$$

Izračunajmo še izraz $e[-1, 1] + f[-1, 1]$. Iz definicij lahko zapišemo

$$e[-1, 1] + f[-1, 1] = \begin{cases} [-e, e] + [-f, f] = (e + f)[-1, 1] & \text{za } e, f \geq 0 \\ [e, -e] + [-f, f] = (-e + f)[-1, 1] & \text{za } e < 0, f \geq 0 \\ [-e, e] + [f, -f] = (e - f)[-1, 1] & \text{za } e \geq 0, f < 0 \\ [e, -e] + [f, -f] = (-e - f)[-1, 1] & \text{za } e, f < 0 \end{cases}.$$

S poenostavitvijo zgornjega izraza lahko krajše zapišemo dobljeno enakost kot

$$e[-1, 1] + f[-1, 1] = (|e| + |f|)[-1, 1]. \quad (2.8)$$

2.3 Vektorski intervali

Vektorski interval $[\mathbf{x}]$ je v splošnem definiran kot poljubna kompaktna množica n -teric, ki ga določajo. Naj bosta $[\mathbf{p}]$ in $[\mathbf{q}]$ vektorska intervala. Vsota $[\mathbf{p}] + [\mathbf{q}]$ je definirana kot

$$[\mathbf{p}] + [\mathbf{q}] = \{ \mathbf{a} + \mathbf{b} \mid \mathbf{a} \in [\mathbf{p}], \mathbf{b} \in [\mathbf{q}] \}. \quad (2.9)$$

Ker je takšna aritmetika precej težavna in v praktični uporabi le redko potrebna, se bomo v tem delu omejili na vektorske intervale, ki jih dobimo kot rezultat tenzorskega produkta dveh ali več skalarnih intervalov.

Naj bodo $a_i, b_i \in \mathbb{R}$, $i = 1, 2, \dots, n$, in naj velja $a_i \leq b_i$. Potem je tenzorski vektorski interval $[\mathbf{p}]$ definiran kot

$$[\mathbf{p}] = [a_1, b_1] \times [a_2, b_2] \times \dots \times [a_n, b_n] = \{ (p_1, p_2, \dots, p_n) \mid p_i \in [a_i, b_i] \}. \quad (2.10)$$

Za $n = 2$ je vektorski interval pravokotnik v ravnini, pri $n = 3$ pa kvader v prostoru. Robovi politopa, ki ga opisuje vektorski interval, so vzporedni koordinatnim osem. Enačba (2.9) se za vektorska intervala $[\mathbf{p}] = [a_1, b_1] \times [a_2, b_2] \times \cdots \times [a_n, b_n]$ in $[\mathbf{q}] = [c_1, d_1] \times [c_2, d_2] \times \cdots \times [c_n, d_n]$ poenostavi v

$$[\mathbf{p}] + [\mathbf{q}] = [a_1 + c_1, b_1 + d_1] \times [a_2 + c_2, b_2 + d_2] \times \cdots \times [a_n + c_n, b_n + d_n].$$

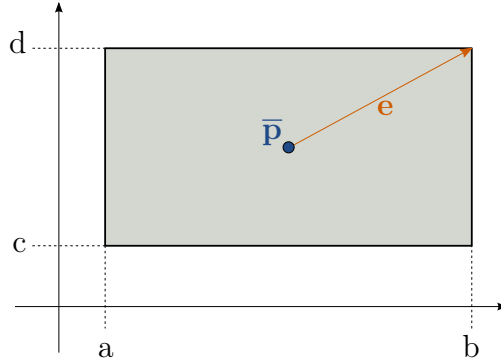
Vektorski interval lahko dobimo tudi kot rezultat množenja n -terice s skalarnim intervalom. Naj bo $\mathbf{a} = (a_1, a_2, \dots, a_n)$ n -terica in $[c, d]$ skalarni interval. Njun produkt je enak

$$\begin{aligned} \mathbf{a}[c, d] &= (a_1, a_2, \dots, a_n)[c, d] \\ &= (a_1[c, d]) \times (a_2[c, d]) \times \cdots \times (a_n[c, d]). \end{aligned}$$

Vektorski interval pa lahko opišemo tudi dugače. Naj bo $[\mathbf{p}]$ enak vektorski interval kot v enačbi (2.10). Iz definicije tenzorskega vektorskega intervala sledi

$$\begin{aligned} [\mathbf{p}] &= [a_1, b_1] \times \cdots \times [a_n, b_n] \\ &= (a_1, \dots, a_n) + [0, b_1 - a_1] \times \cdots \times [0, b_n - a_n] \\ &= (a_1, \dots, a_n) + (b_1 - a_1, \dots, b_n - a_n)[0, 1] \\ &= (a_1, \dots, a_n) + \frac{1}{2}(b_1 - a_1, \dots, b_n - a_n)[0, 2] \\ &= \frac{1}{2}(a_1 + b_1, \dots, a_n + b_n) + \frac{1}{2}(b_1 - a_1, \dots, b_n - a_n)[-1, 1] \\ &= \bar{\mathbf{p}} + \mathbf{e}[i], \end{aligned}$$

kjer $\bar{\mathbf{p}} = \frac{1}{2}(a_1 + b_1, \dots, a_n + b_n)$ predstavlja središče politopa, $\mathbf{e} = \frac{1}{2}(b_1 - a_1, \dots, b_n - a_n)$ predstavlja razdalje do robnih ploskev politopa, ter je $[i] = [-1, 1]$. Takšen zapis bomo imenovali usredinjeni zapis vektorskega intervala. Na sliki 2.1 je shematsko prikazan vektorski interval $[a, b] \times [c, d]$, na katerem sta označena $\bar{\mathbf{p}}$ in \mathbf{e} .

Slika 2.1: Vektorski interval $[a, b] \times [c, d]$.

2.4 Intervalski polinomi

Intervalski polinom je polinom, katerega koeficienti so intervali. Intervalske polinome bomo označevali z $[p](t)$. Ker bomo intervalske polinome potrebovali pri definiciji in analizi Bézierovih intervalskih krivulj, si oglejmo zapis intervalskega polinoma v Bernsteinovi bazi na intervalu $[0, 1]$.

Vsak intervalski polinom $[p](t)$ stopnje n lahko zapišemo kot

$$[p](t) = \sum_{i=0}^n [u_i] B_i^n(t),$$

kjer je $[u_i] = [a_i, b_i]$ za $i \in \{0, 1, \dots, n\}$. Ker za $t \in [0, 1]$ velja $B_i^n(t) \geq 0$, lahko zapišemo $[p](t)$ kot

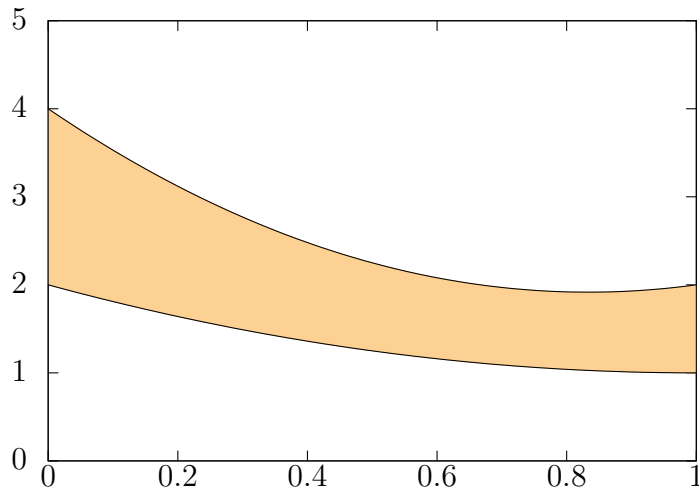
$$\begin{aligned} [p](t) &= \sum_{i=0}^n [a_i, b_i] B_i^n(t) \\ &= \sum_{i=0}^n [a_i B_i^n(t), b_i B_i^n(t)] \\ &= \left[\sum_{i=0}^n a_i B_i^n(t), \sum_{i=0}^n b_i B_i^n(t) \right] \\ &= [p_{\min}(t), p_{\max}(t)], \end{aligned}$$

kjer sta $p_{\min}(t)$ in $p_{\max}(t)$ navadna polinoma stopnje n . Drugi korak sledi iz (2.6), tretji korak pa je posplošitev definicije seštevanja na več kot dva intervala.

Oglejmo si še sliko 2.2, kjer je narisani intervalski polinom

$$[p](t) = [1, 2]t^2 + [2, 3]t(1 - t) + [2, 4](1 - t)^2. \quad (2.11)$$

Osenčeno območje na sliki imenujemo *širina polinoma* in se ponavadi uporablja kot mera za določitev tolerance aproksimacije z intervalskimi polinomi.



Slika 2.2: Prikaz intervalskega polinoma iz enačbe (2.11).

Poglavje 3

Intervalske Bézierove krivulje

V tem poglavju se bomo posvetili definiciji intervalskih Bézierovih krivulj ter opisu osnovnih algoritmov, ki jih potrebujemo pri delu z intervalskimi Bézierovimi krivuljami.

3.1 Definicija intervalske Bézierove krivulje

Osnova intervalske Bézierove krivulje je navadna (neintervalska) Bézierova krivulja, ki jo geometrijsko-fizikalno lahko opišemo na naslednji način [7].

Naj bodo $m_i, i = 0, 1, \dots, n$, mase teles v prostoru. Položaje teles označimo s $\mathbf{P}_i \in \mathbb{R}^3, i = 0, 1, \dots, n$ (slika 3.1a). Težišče teles lahko izračunamo kot

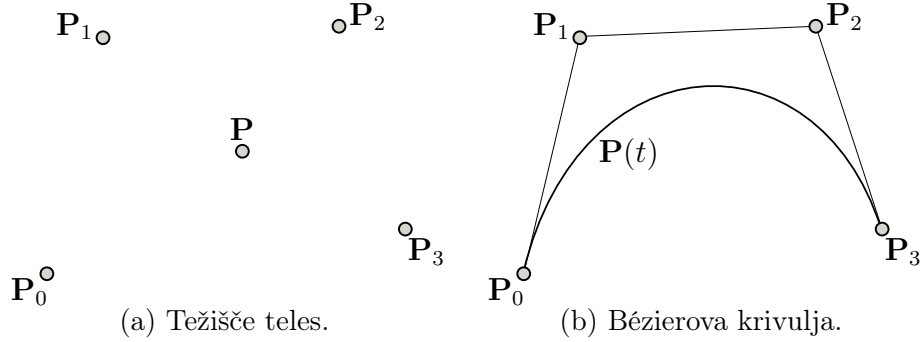
$$\mathbf{P} = \frac{\sum_{i=0}^n \mathbf{P}_i m_i}{\sum_{i=0}^n m_i}. \quad (3.1)$$

Če v enačbi (3.1) konstantne mase teles m_i zamenjamo z Bernsteinovimi baznimi polinomi ($m_i = B_i^n(t)$), dobimo parametrično krivuljo

$$\mathbf{P}(t) = \frac{\sum_{i=0}^n \mathbf{P}_i B_i^n(t)}{\sum_{i=0}^n B_i^n(t)}. \quad (3.2)$$

Za $t \in [0, 1]$ velja $\sum_{i=0}^n B_i^n(t) = 1$, kar enačbo (3.2) na intervalu $[0, 1]$ poenostavi v

$$\mathbf{P}(t) = \sum_{i=0}^n \mathbf{P}_i B_i^n(t). \quad (3.3)$$



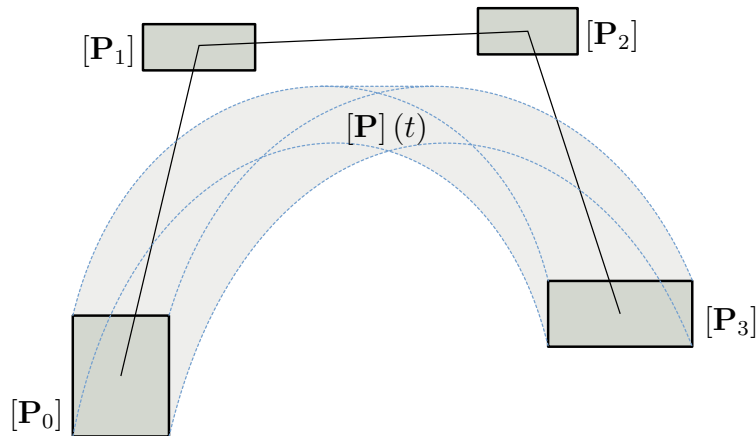
Slika 3.1: Skica težišča teles ter Bézierova krivulja, ki jo dobimo, ko parameter t preteče interval $[0, 1]$.

Krivuljo, ki jo dobimo, ko t preteče interval $[0, 1]$, imenujemo Bézierova krivulja stopnje n , točke P_i pa Bézierove kontrolne točke (slika 3.1b).

Intervalsko Bézierovo krivuljo dobimo, če kontrolne točke P_i zamenjamo z vektorskimi intervali $[P_i]$, kar si lahko predstavljamo kot napako pri določanju položaja teles [8]. Intervalsko Bézierovo krivuljo bomo označevali z $[P](t)$, matematično pa je definirana kot

$$[P](t) = \sum_{i=0}^n [P_i] B_i^n(t). \quad (3.4)$$

Grafično si lahko intervalsko Bézierovo krivuljo predstavljamo kot osenčen del na sliki 3.2.



Slika 3.2: Skica intervalske Bézierove krivulje.

Če v enačbi (3.4) vektorske intervale zapišemo v usredinjeni obliki, dobimo zapis intervalske Bézierove krivulje

$$\begin{aligned}
 [\mathbf{P}](t) &= \sum_{k=0}^n [\mathbf{P}_k] B_k^n(t) \\
 &= \sum_{k=0}^n (\bar{\mathbf{P}}_k + \mathbf{e}_k[i]) B_k^n(t) \\
 &= \sum_{k=0}^n \bar{\mathbf{P}}_k B_k^n(t) + \sum_{k=0}^n \mathbf{e}_k B_k^n(t)[i] \\
 &= \sum_{k=0}^n \bar{\mathbf{P}}_k B_k^n(t) + [i] \sum_{k=0}^n \mathbf{e}_k B_k^n(t) \\
 &= \bar{\mathbf{P}}(t) + [i]\mathbf{e}(t),
 \end{aligned}$$

kjer je $\bar{\mathbf{P}}(t) = \sum_{k=0}^n \bar{\mathbf{P}}_k B_k^n(t)$ in $\mathbf{e}(t) = \sum_{k=0}^n \mathbf{e}_k B_k^n(t)$. Interval $[i]$ smo v tretjem koraku lahko izpostavili, ker za $k = 0, 1, \dots, n$ in $t \in [0, 1]$ velja, da so vse komponente vektorja \mathbf{e}_k nenegativne, prav tako pa je nenegativen tudi $B_k^n(t)$. Intervalsko Bézierovo krivuljo si tako lahko predstavljamo kot navadno Bézierovo krivuljo $\bar{\mathbf{P}}(t)$, ki ima kontrolne točke v središčih vektorskih intervalov, ter krivuljo napake $\mathbf{e}(t)$.

3.1.1 Afine transformacije parametra

Pri izpeljavi enačbe navadne Bézierove krivulje (3.3) smo za definicijsko območje parametra vzeli interval $[0, 1]$. V nekaterih okoliščinah pa nam to definicijsko območje ne ustreza (na primer, če bi želeli stakniti skupaj več Bézierovih krivulj) in bi želeli za definicijsko območje parametra uporabiti poljuben interval $[a, b]$. V tem primeru uvedemo linearno parametrizacijo $t = \frac{b-u}{b-a}$ in dobimo

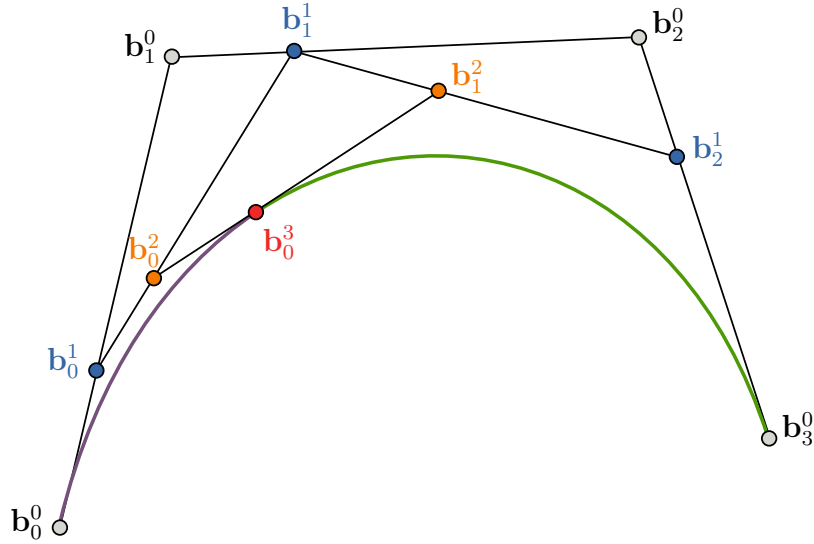
$$\mathbf{P}_{[0,1]}(t) = \mathbf{P}_{[a,b]}(u) = \sum_{i=0}^n \mathbf{P}_i B_i^n\left(\frac{b-u}{b-a}\right),$$

kjer zapis $\mathbf{P}_{[a,b]}(t)$ pomeni Bézierovo krivuljo, kjer je definicijsko območje parametra interval $[a, b]$, $\mathbf{P}(t)$ pa je krajši zapis za $\mathbf{P}_{[0,1]}(t)$ [1, 2].

3.2 De Casteljaujev algoritem

De Casteljaujev algoritem opisuje postopek subdivizije Bézierove krivulje $\mathbf{P}_{[0,1]}(t)$ v dve krivulji $\mathbf{P}_{[0,u]}(t)$ in $\mathbf{P}_{[u,1]}(t)$, katerih unija je prvotna krivulja, obenem pa omogoča stabilen izračun poljubne točke na krivulji.

Geometrijsko je de Casteljaujev algoritem zaporedje izračunov točk, ki izbrano daljico delijo v razmerju $u : (1 - u)$. Na sliki 3.3 je grafično prikazano izvajanje algoritma za krivuljo stopnje 3 pri vrednosti parametra $t = \frac{1}{3}$, celoten postopek za krivuljo poljubne stopnje in pri poljubni vrednosti parametra pa je predstavljen v algoritmu 3.1.



Slika 3.3: Grafični prikaz de Casteljaujevega algoritma.

Algoritem 3.1 De Casteljaujev algoritem.

- 1: **for** $i = 0$ to n **do**
 - 2: $\mathbf{b}_i^0 \leftarrow \mathbf{P}_i$
 - 3: **for** $j = 1$ to n **do**
 - 4: **for** $i = 0$ to $n - j$ **do**
 - 5: $\mathbf{b}_i^j \leftarrow (1 - u)\mathbf{b}_i^{j-1} + u\mathbf{b}_{i+1}^{j-1}$
 - 6: $\mathbf{P}(u) \leftarrow \mathbf{b}_0^n$
-

Če je vhodna krivulja za algoritem oblike $\mathbf{P}(t) = \sum_{i=0}^n \mathbf{P}_i B_i^n(t)$, potem sta Bézierovi krivulji, ki ju dobimo s subdivizijo prvotne krivulje pri vrednosti parametra $t = u$, enaki $\mathbf{P}_1(t) = \sum_{i=0}^n \mathbf{b}_0^i B_i^n(t)$ in $\mathbf{P}_2(t) = \sum_{i=0}^n \mathbf{b}_i^{n-i} B_i^n(t)$, obenem pa točka $\mathbf{b}_0^n = \mathbf{P}(u)$ leži na Bézierovi krivulji. Točke kontrolnega poligona vijolične krivulje na sliki 3.3 so tako \mathbf{b}_0^0 , \mathbf{b}_0^1 , \mathbf{b}_0^2 in \mathbf{b}_0^3 , točke kontrolnega poligona zelene krivulje pa \mathbf{b}_0^3 , \mathbf{b}_1^2 , \mathbf{b}_2^1 in \mathbf{b}_3^0 .

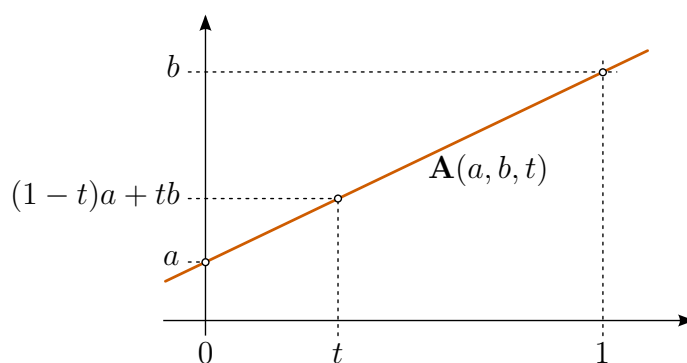
Algoritem lahko posplošimo na intervalske Bézierove krivulje na enak način kot smo posplošili Bézierove krivulje: kontrolne točke zamenjamo z vektorskimi intervali, pri izračunih pa uporabljamo pravila intervalske aritmetike, ki smo jih opisali v poglavju 2. Preden pa opravimo posplošitev, si oglejmo nekaj podrobnosti, na katere moramo biti pozorni pri prenosu algoritma na intervalsko aritmetiko.

3.2.1 Afine preslikave intervalov

De Casteljauev algoritem je v svoji osnovi le zaporedje afinih preslikav dveh točk a in b

$$\mathbf{A}(a, b, t) = (1 - t)a + tb, \quad (3.5)$$

kjer so $a, b, t \in \mathbb{R}$. Rezultat te preslikave bo vedno točka, grafično pa lahko prikažemo takšno preslikavo kot premico, ki poteka skozi točki $(0, a)$ in $(1, b)$ (slika 3.4).



Slika 3.4: Afina preslikava dveh točk.

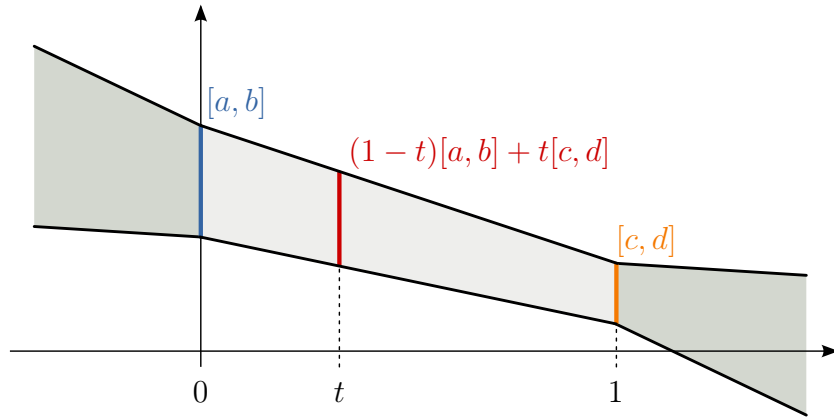
Afino preslikavo intervalov $[a, b]$ in $[c, d]$ pa zapišemo kot

$$\begin{aligned} [\mathbf{A}]([a, b], [c, d], t) &= (1 - t)[a, b] + t[c, d] \\ &= \{ (1 - t)u + tv \mid u \in [a, b], v \in [c, d] \}. \end{aligned} \quad (3.6)$$

Opazimo, da je rezultat afine preslikave dveh intervalov zopet interval, kaj več pa iz enačbe (3.6) ne moremo povedati o naravi tega intervala. Enačbo (3.6) pa lahko z upoštevanjem enačb (2.6) in (2.7) prepišemo v

$$[\mathbf{A}]([a, b], [c, d], t) = \begin{cases} [(1 - t)a + td, (1 - t)b + tc] & \text{za } t < 0 \\ [(1 - t)a + tc, (1 - t)b + td] & \text{za } 0 \leq t \leq 1 \\ [(1 - t)b + tc, (1 - t)a + td] & \text{za } t > 1 \end{cases}$$

in grafično prikažemo na sliki 3.5 kot osenčen del. Iz slike je jasno razvidno, da je širina intervala $[\mathbf{A}]([a, b], [c, d], t)$ za $t \in [0, 1]$ omejena in med širinama intervalov $[a, b]$ ter $[c, d]$ (svetlo osenčen del na sliki 3.5), za $t \notin [0, 1]$ pa širina intervala narašča linearno v neskončnost (temno osenčen del).

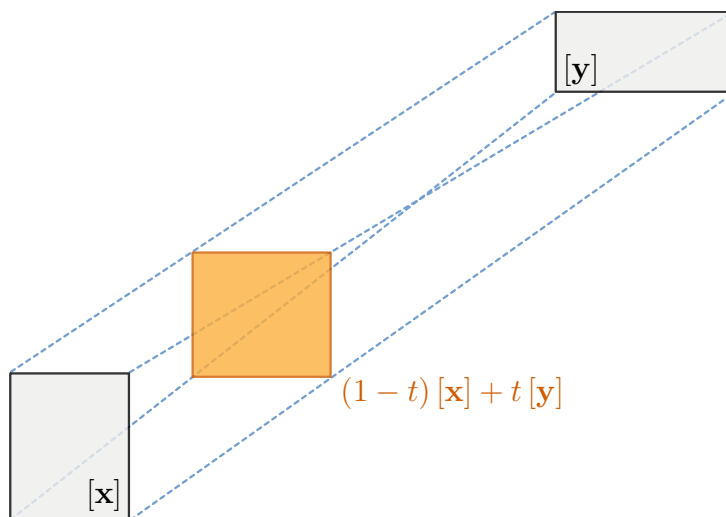


Slika 3.5: Afina preslikava dveh intervalov.

Podobno lahko afino preslikavo vektorskih intervalov $[\mathbf{a}]$ in $[\mathbf{b}]$ zapišemo kot

$$\begin{aligned} [\mathbf{A}]([\mathbf{a}], [\mathbf{b}], t) &= (1 - t)[\mathbf{a}] + t[\mathbf{b}] \\ &= \{ (1 - t)\mathbf{a} + t\mathbf{b} \mid \mathbf{a} \in [\mathbf{a}], \mathbf{b} \in [\mathbf{b}] \}, \end{aligned} \quad (3.7)$$

kjer dejansko izvajamo afine preslikave navadnih intervalov za vsako dimenzijo politopa, ki ga opisuje vektorski interval. Na sliki 3.6 je grafično prikazan primer za dvodimenzionalna vektorska intervala $[\mathbf{x}]$ in $[\mathbf{y}]$ ter $t \in [0, 1]$.



Slika 3.6: Afina preslikava dveh vektorskih intervalov.

Afina preslikavo med dvema vektorskima intervaloma pa lahko zapišemo tudi v usredinjeni obliki. Naj bo $[\mathbf{A}]$ afina preslikava, $[\mathbf{v}] = \bar{\mathbf{v}} + \mathbf{e}_v[i]$ in $[\mathbf{u}] = \bar{\mathbf{u}} + \mathbf{e}_u[i]$ pa vektorska intervala. Enačba (3.7) se poenostavi v

$$\begin{aligned}
 [\mathbf{A}]([\mathbf{v}], [\mathbf{u}], t) &= (1-t)[\mathbf{v}] + t[\mathbf{u}] \\
 &= (1-t)(\bar{\mathbf{v}} + \mathbf{e}_v[i]) + t(\bar{\mathbf{u}} + \mathbf{e}_u[i]) \\
 &= (1-t)\bar{\mathbf{v}} + t\bar{\mathbf{u}} + (1-t)\mathbf{e}_v[i] + t\mathbf{e}_u[i] \\
 &= (1-t)\bar{\mathbf{v}} + t\bar{\mathbf{u}} + (|1-t|\mathbf{e}_v + |t|\bar{\mathbf{e}}_u)[i],
 \end{aligned}$$

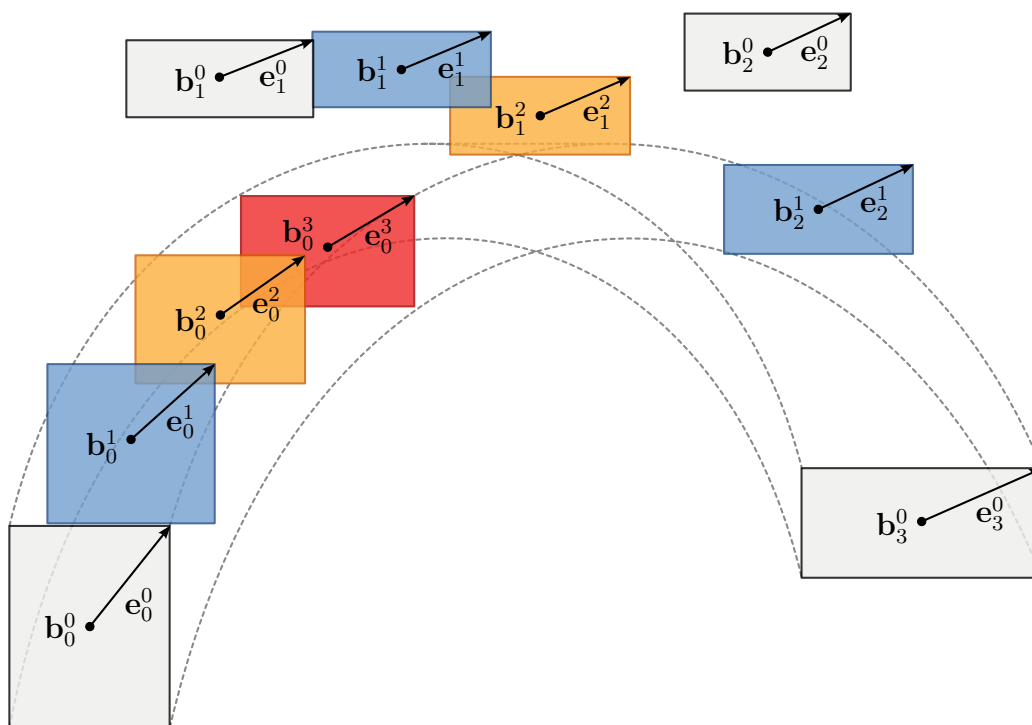
kjer zadnji korak sledi iz enakosti (2.8). Če sedaj vrednosti parametra omejimo na $t \in [0, 1]$, lahko nadalje poenostavimo

$$\begin{aligned}
 (1-t)\bar{\mathbf{v}} + t\bar{\mathbf{u}} + (|1-t|\mathbf{e}_v + |t|\bar{\mathbf{e}}_u)[i] &= (1-t)\bar{\mathbf{v}} + t\bar{\mathbf{u}} + ((1-t)\mathbf{e}_v + t\bar{\mathbf{e}}_u)[i] \\
 &= \mathbf{A}(\bar{\mathbf{v}}, \bar{\mathbf{u}}, t) + \mathbf{A}(\mathbf{e}_v, \mathbf{e}_u, t)[i],
 \end{aligned}$$

kar dejansko pomeni, da lahko afino preslikavo dveh vektorskih intervalov poiščemo z izračunom afinih preslikav središč intervalov in napake.

3.2.2 Intervalski de Casteljaujev algoritem

V prejšnjem razdelku smo ugotovili, da lahko za vrednosti parametra $t \in [0, 1]$ varno¹ izvajamo afino transformacijo, ki je osnova de Casteljaujevega algoritma. S tem znanjem lahko de Casteljaujev algoritem varno posplošimo. Zapisan v usredinjeni obliki je predstavljen v algoritmu 3.2, grafični prikaz postopka za krivuljo stopnje 3 pa je na sliki 3.7.



Slika 3.7: Grafični prikaz intervalskega de Casteljaujevega algoritma.

3.3 Višanje stopnje Bézierove krivulje

Postopka višanja stopnje Bézierove krivulje se poslužimo predvsem kadar želimo pretvoriti informacijo iz ene oblike v drugo oziroma poenotiti zapis brez spreminjanja oblike (na primer, vse krivulje zapisati kot krivulje stopnje k).

¹Varno tu pomeni, da širine intervalov, ki jih bomo dobili kot rezultat, ne bodo nenadzorovano rasle v neskočnost.

Algoritem 3.2 Intervalski de Casteljaujev algoritem.

```

1: for  $i = 0$  to  $n$  do
2:    $\mathbf{b}_i^0 \leftarrow \text{CENTER}([\mathbf{P}_i])$  ▷ Izračun središča intervala
3:    $\mathbf{e}_i^0 \leftarrow \text{ERROR}([\mathbf{P}_i])$  ▷ Izračun razdalj do mejnih ploskev
4: for  $j = 1$  to  $n$  do
5:   for  $i = 0$  to  $n - j$  do
6:      $\mathbf{b}_i^j \leftarrow (1 - u)\mathbf{b}_i^{j-1} + u\mathbf{b}_{i-1}^{j-1}$ 
7:      $\mathbf{e}_i^j \leftarrow (1 - u)\mathbf{e}_i^{j-1} + u\mathbf{e}_{i-1}^{j-1}$ 
8:  $[\mathbf{P}](u) \leftarrow \text{INTERVAL}(\mathbf{b}_0^n, \mathbf{e}_0^n)$  ▷ Usredinjena  $\rightarrow$  navadna oblika

```

Pa si najprej oglejmo algoritem za navadne Bézierove krivulje, ki ga bomo potem z nekaj previdnosti posplošili na intervalske Bézierove krivulje.

3.3.1 Višanje stopnje navadne krivulje

Naj bo $\mathbf{P}(t) = \sum_{i=0}^n \mathbf{P}_i B_i^n(t)$ Bézierova krivulja stopnje n , ki ji želimo stopnjo povečati na $n+1$, in $\tilde{\mathbf{P}}(t) = \sum_{i=0}^{n+1} \tilde{\mathbf{P}}_i B_i^{n+1}(t)$ krivulja, ki jo dobimo kot rezultat višanja stopnje. Veljati mora $\mathbf{P}(t) = \tilde{\mathbf{P}}(t)$ oziroma

$$\sum_{i=0}^n \mathbf{P}_i B_i^n(t) = \sum_{i=0}^{n+1} \tilde{\mathbf{P}}_i B_i^{n+1}(t). \quad (3.8)$$

Če levo stran v enačbi (3.8) pomnožimo z $1 = (1 - t) + t$, dobimo

$$\begin{aligned} ((1 - t) + t) \sum_{i=0}^n \mathbf{P}_i B_i^n(t) &= \sum_{i=0}^n \mathbf{P}_i ((1 - t)B_i^n(t) + tB_i^n(t)) \\ &= \sum_{i=0}^n \mathbf{P}_i \left(\frac{n+1-i}{n+1} B_i^{n+1}(t) + \frac{i+1}{n+1} B_{i+1}^{n+1}(t) \right), \end{aligned}$$

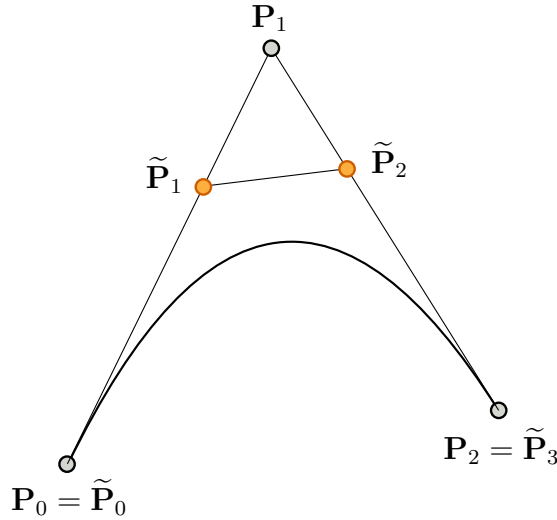
kjer smo v drugem koraku upoštevali enakosti (2.3) in (2.4). Ko člene v vsoti združimo glede na Bernsteinove bazne polinome, dobimo

$$\mathbf{P}_0 B_0^{n+1}(t) + \sum_{i=1}^n \left(\frac{i}{n+1} \mathbf{P}_{i-1} + \frac{n+1-i}{n+1} \mathbf{P}_i \right) B_i^n(t) + \mathbf{P}_n B_{n+1}^{n+1}(t). \quad (3.9)$$

Iz enačb (3.8) in (3.9) lahko s primerjanjem kontrolnih točk pri enakih koeficientih določimo nove kontrolne točke $\tilde{\mathbf{P}}_i$, $i = 0, 1, \dots, n + 1$, kot

$$\begin{aligned}\tilde{\mathbf{P}}_0 &= \mathbf{P}_0 \\ \tilde{\mathbf{P}}_i &= \alpha_i \mathbf{P}_{i-1} + (1 - \alpha_i) \mathbf{P}_i, \quad i = 1, 2, \dots, n, \\ \tilde{\mathbf{P}}_{n+1} &= \mathbf{P}_n,\end{aligned}\tag{3.10}$$

kjer je $\alpha_i = \frac{i}{n+1}$. Na sliki 3.8 je prikazan primer višanja stopnje kvadratične krivulje v kubično, algoritem 3.3 pa prikazuje psevdokodo postopka za krivuljo poljubne stopnje.



Slika 3.8: Višanje stopnje kvadratične Bezierove krivulje.

Algoritem 3.3 Višanje stopnje navadne krivulje.

- 1: $\tilde{\mathbf{P}}_0 \leftarrow \mathbf{P}_0$
 - 2: **for** $j = 1$ to n **do**
 - 3: $\alpha \leftarrow \frac{j}{n+1}$
 - 4: $\tilde{\mathbf{P}}_j \leftarrow \alpha \mathbf{P}_{j-1} + (1 - \alpha) \mathbf{P}_j$
 - 5: $\tilde{\mathbf{P}}_{n+1} \leftarrow \mathbf{P}_n$
-

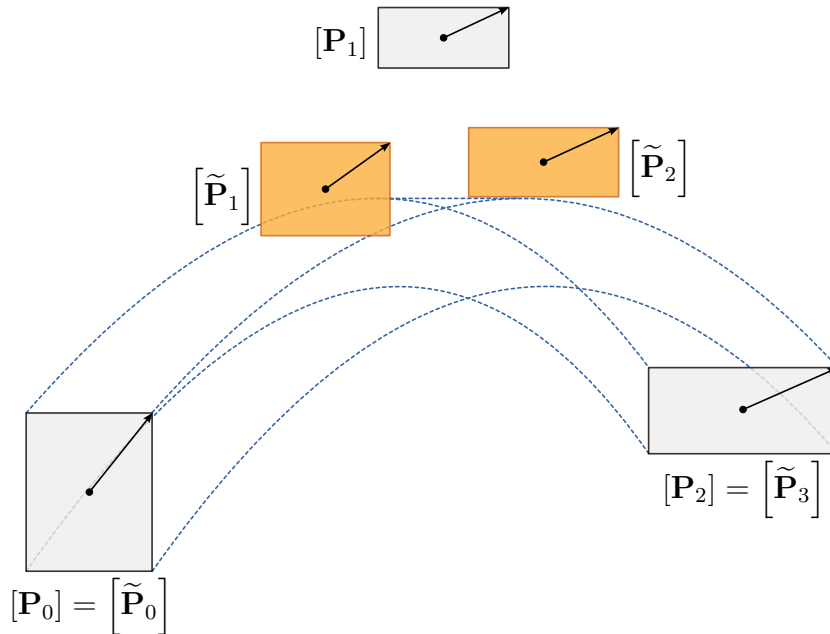
3.3.2 Višanje stopnje intervalske krivulje

Posplošitev je v tem primeru enostavna, ker je iz enačbe (3.10) razvidno, da novo kontrolno točko dobimo z linearno kombinacijo dveh starih zaporednih kontrolnih točk. Za takšne vrste preslikav smo v razdelku 3.2.1 pokazali, da slikajo omejene vektorske intervale v omejene vektorske intervale, s čimer imamo izpolnjene pogoje za uspešno posplošitev.

Naj bo $[\mathbf{P}](t)$ intervalska Bézierova krivulja s kontrolnimi točkami $[\mathbf{P}_k] = \bar{\mathbf{P}}_k + \mathbf{e}_k[i]$, $k = 0, 1, \dots, n$. Kontrolne točke intervalske krivulje $[\tilde{\mathbf{P}}](t)$, ki jo dobimo z višanjem stopnje krivulje $[\mathbf{P}](t)$, so enake

$$\begin{aligned} [\tilde{\mathbf{P}}_0] &= [\mathbf{P}_0], \\ [\tilde{\mathbf{P}}_i] &= \alpha_i [\mathbf{P}_{i-1}] + (1 - \alpha_i) [\mathbf{P}_i], \quad i = 1, 2, \dots, n, \\ [\tilde{\mathbf{P}}_{n+1}] &= [\mathbf{P}_n], \end{aligned}$$

kjer je $\alpha_i = \frac{i}{n+1}$ [3]. Izračun novih kontrolnih točk v usredinjeni obliki je predstavljen v algoritmu 3.4.



Slika 3.9: Višanje stopnje intervalske Bezierove krivulje.

Algoritem 3.4 Višanje stopnje intervalske krivulje.

```

1: for  $i = 0$  to  $n$  do
2:    $\mathbf{c}_i \leftarrow \text{CENTER}([\mathbf{P}_i])$ 
3:    $\mathbf{e}_i \leftarrow \text{ERROR}([\mathbf{P}_i])$ 
4:  $[\tilde{\mathbf{P}}_0] \leftarrow [\mathbf{P}_0]$ 
5: for  $j = 1$  to  $n$  do
6:    $\alpha \leftarrow \frac{i}{n+1}$ 
7:    $\mathbf{c} \leftarrow \alpha \mathbf{c}_{i-1} + (1 - \alpha) \mathbf{c}_i$ 
8:    $\mathbf{e} \leftarrow \alpha \mathbf{e}_{i-1} + (1 - \alpha) \mathbf{e}_i$ 
9:    $[\tilde{\mathbf{P}}_i] \leftarrow \text{INTERVAL}(\mathbf{c}, \mathbf{e})$ 
10:  $[\tilde{\mathbf{P}}_{n+1}] \leftarrow [\mathbf{P}_n]$ 

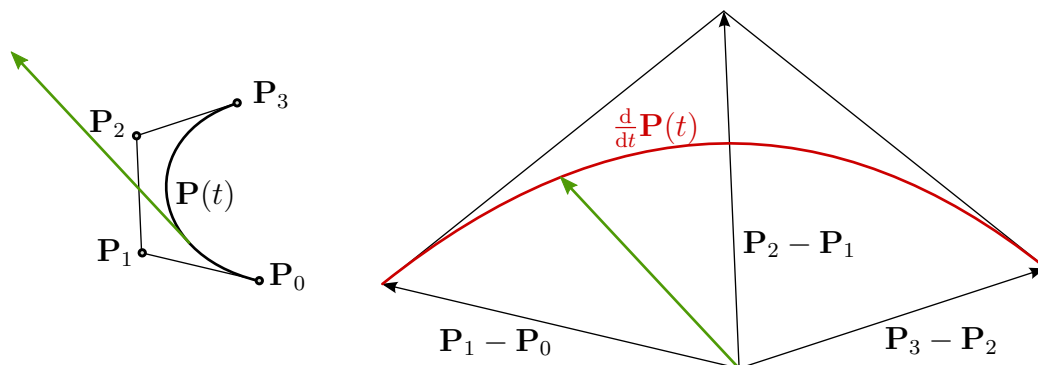
```

3.4 Odvod Bézierove krivulje

Pri izrisu intervalske Bézierove krivulje bomo potrebovali tudi njen prvi in drugi odvod, zato si sedaj pogledjmo, kako ju izračunati. Naj bo $\mathbf{P}(t) = \sum_{i=0}^n \mathbf{P}_i B_i^n(t)$ Bézierova krivulja. Njen odvod je enak

$$\begin{aligned}
\frac{d}{dt} \mathbf{P}(t) &= \frac{d}{dt} \sum_{i=0}^n \mathbf{P}_i B_i^n(t) \\
&= \sum_{i=0}^n \mathbf{P}_i \frac{d}{dt} B_i^n(t) \\
&= \sum_{i=0}^n \mathbf{P}_i (n B_{i-1}^{n-1}(t) - n B_i^{n-1}(t)) \\
&= n \sum_{i=1}^n \mathbf{P}_i B_{i-1}^{n-1}(t) - n \sum_{i=0}^{n-1} \mathbf{P}_i B_i^{n-1}(t) \\
&= n \sum_{i=0}^{n-1} \mathbf{P}_{i+1} B_i^{n-1}(t) - n \sum_{i=0}^{n-1} \mathbf{P}_i B_i^{n-1}(t) \\
&= n \sum_{i=0}^{n-1} (\mathbf{P}_{i+1} - \mathbf{P}_i) B_i^{n-1}(t),
\end{aligned}$$

kar je zopet Bézierova krivulja, katere kontrolne točke so difference kontrolnih točk prvotne krivulje. Grafično je odvod predstavljen na sliki 3.10, kjer zeleno obarvan vektor predstavlja smer in velikost odvoda v točki $t = 0, 25$.



Slika 3.10: Kubična Bézierova krivulja (levo) in njen odvod (desno).

Posplošitev odvoda na intervalske Bézierove krivulje je trivialna: kontrolne točke zamenjamo s kontrolnimi intervali in smo končali.

3.5 Izračun večih točk na Bézierovi krivulji

Večina medijev, na katerih prikazujemo krivulje, potrebuje za prikaz slike gosto množico točk, ki jih nato po potrebi poveže z ravnimi odseki. V naslednjih nekaj odstavkih si bomo ogledali tri metode, s katerimi lahko izračunamo takšno množico točk. Pri vseh izračunih bomo predpostavili, da želimo izračunati vrednosti $[P](t_i)$, kjer je $t_i = \frac{i}{m}$, $i = 0, 1, \dots, m$, in $m \in \mathbb{N}$.

3.5.1 Izračun s pomočjo de Casteljaujevega algoritma

Osnovna ideja te metode je, da izračun $[P](t_i)$ opravimo s pomočjo de Casteljaujevega algoritma. Če predpostavimo, da sta $\mathbf{b}, \mathbf{e} \in \mathbb{R}^3$, vrstici 6 in 7 v algoritmu 3.2 opravita vsaka po 6 množenj in 4 seštevanja, število ponovitev teh vrstic pa je enako

$$\sum_{j=1}^n (n - j + 1) = \frac{n(n+1)}{2}.$$

Skupaj torej za izračun vseh vrednosti izvedemo $O(mn^2)$ operacij.

3.5.2 Hornerjev algoritem v Bernsteinovi bazi

Najbolj učinkovita metoda izračuna vrednosti polinoma je uporaba Hornerjevega algoritma. Pa pogledjmo, kako bi prilagodili ta algoritem za polinome, zapisane v Bernsteinovi bazi. Polinom $p(t)$ lahko preoblikujemo v

$$\begin{aligned} p(t) &= \sum_{i=0}^n a_i B_i^n(t) \\ &= \binom{n}{0} a_0 (1-t)^n + \binom{n}{1} a_1 t (1-t)^{n-1} + \cdots + \binom{n}{n} a_n t^n \\ &= \left(\left(\cdots (z_0 (1-t) + z_1) (1-t) + \cdots + z_{n-2} \right) (1-t) + z_{n-1} \right) (1-t) + z_n, \end{aligned}$$

kjer smo z z_i označili člen $\binom{n}{i} a_i t^i$. Če pri izračunu zgornjega izraza upoštevamo, da za $n, i \in \mathbb{N}$, $n \geq i > 0$, velja

$$\binom{n}{i} = \frac{n-i+1}{i} \binom{n}{i-1},$$

dobimo zelo učinkovit postopek za izračun vrednosti polinoma, ki je opisan v algoritmu 3.5. Takšen izračun potrebuje $9n - 6$ množenj, $5n - 2$ seštevanj, eno odštevanje in $n - 1$ deljenj. Za izris intervalske krivulje $[\mathbf{P}](t)$ moramo ta postopek uporabiti dvakrat (za izračun središč intervalov in za izračun napak). Časovna zahtevnost tega algoritma je torej $O(mn)$.

3.5.3 Metoda premih končnih diferenc

Prej opisani metodi sta izračun vsake točke opravili neodvisno in nista uporabili dejstva, da so razlike med vrednostmi parametra t , v katerih želimo izračunati vrednost polinoma, enake. Metoda premih končnih diferenc pa rezultate prejšnjega izračuna uporabi kot osnovo za izračun trenutne vrednosti.

Naj bo $p(t)$ polinom stopnje n , ki ga želimo izračunati v točkah t_i . Defini-

Algoritem 3.5 Hornerjev algoritem v Bernsteinovi bazi.

```

1:  $u \leftarrow 1 - t$ 
2:  $d \leftarrow 1$  ▷ Potenca parametra  $t$ 
3:  $b \leftarrow 1$  ▷ Vrednost binomskega simbola
4:  $\mathbf{v} \leftarrow \mathbf{a}_0 \cdot u$  ▷ Izračunana vrednost
5: for  $i = 1$  to  $n - 1$  do
6:    $d \leftarrow d \cdot t$ 
7:    $b \leftarrow b \cdot (n - i + 1) / i$ 
8:    $\mathbf{v} \leftarrow (\mathbf{v} + \mathbf{a}_i \cdot (b \cdot t))u$ 
9:  $\mathbf{v} \leftarrow \mathbf{v} + \mathbf{a}_n \cdot (t \cdot d)$ 

```

rajmo j -to premo končno diferenco kot

$$\Delta_j(t) = \begin{cases} p(t + \delta) - p(t) & j = 1 \\ \Delta_{j-1}(t + \delta) - \Delta_{j-1}(t) & \text{sicer} \end{cases},$$

kjer je $\delta = t_{i+1} - t_i$. S preprostim izračunom lahko pokažemo, da je $\Delta_1(t)$ polinom stopnje največ $n - 1$. Po definiciji je

$$\begin{aligned} \Delta_1(t) &= p(t + \delta) - p(t) \\ &= \sum_{i=0}^n a_i (t + \delta)^i - \sum_{i=0}^n a_i t^i \\ &= \sum_{i=0}^n \sum_{k=0}^i a_i \binom{i}{k} t^k \delta^{i-k} - \sum_{i=0}^n a_i t^i. \end{aligned}$$

Člen iz prve vsote, ki vsebuje t^n , dobimo pri $i = k = n$ in je enak $a_n t^n$. Enak člen dobimo v drugi vsoti, kar pomeni, da razlika vsebuje člene stopnje največ $n - 1$. Posledično bo n -ta diferenca polinom stopnje 0 oziroma konstanta, vse nadaljne difference pa bodo enake 0.

Zgornje ugotovitve lahko uporabimo za izračun vrednosti polinoma $p(t)$ v točkah t_i po metodi, ki je predstavljena v algoritmu 3.6. Prvi del metode izračuna začetne vrednosti (v tabeli 3.1 so prikazane z modro barvo), s pomočjo katerih v drugem delu le s seštevanjem dobimo vse preostale vrednosti

(v tabeli 3.1 prikazane z črno barvo). Izračunane vrednosti se nahajajo v seznamu $v_{0,i}$, $i = 0, 1, \dots, m$.

| | | | | | | |
|-----------------|---------------------|---------------------|---------------------|---------------------|--------------|---------|
| $p(t_i)$ | $p(t_{i+1})$ | $p(t_{i+2})$ | $p(t_{i+3})$ | $p(t_{i+4})$ | $p(t_{i+5})$ | \dots |
| $\Delta_1(t_i)$ | $\Delta_1(t_{i+1})$ | $\Delta_1(t_{i+2})$ | $\Delta_1(t_{i+3})$ | $\Delta_1(t_{i+4})$ | \dots | |
| $\Delta_2(t_i)$ | $\Delta_2(t_{i+1})$ | $\Delta_2(t_{i+2})$ | $\Delta_2(t_{i+3})$ | \dots | | |
| $\Delta_3(t_i)$ | $\Delta_3(t_{i+1})$ | $\Delta_3(t_{i+2})$ | \dots | | | |

Tabela 3.1: Tabela diferenc za polinom stopnje 3.

Algoritem 3.6 Metoda diferenc.

| | |
|--|--------------------------------|
| 1: for $i = 0$ to n do | ▷ Izračun začetnih vrednosti |
| 2: $\mathbf{v}_{0,i} \leftarrow \text{EVAL}(p, t_i)$ | ▷ Izračun vrednosti polinoma |
| 3: for $j = 1$ to i do | |
| 4: $\mathbf{v}_{j,i-j} \leftarrow \mathbf{v}_{j-1,i} - \mathbf{v}_{j-1,i-1}$ | |
| 5: for $i = n + 1$ to m do | ▷ Izračun preostalih vrednosti |
| 6: $\mathbf{v}_{n,i-n} \leftarrow \mathbf{v}_{n,i-n-1}$ | |
| 7: for $j = n - 1$ to 0 do | |
| 8: $\mathbf{v}_{j,i-j} \leftarrow \mathbf{v}_{j,i-j-1} - \mathbf{v}_{j+1,j-i-1}$ | |

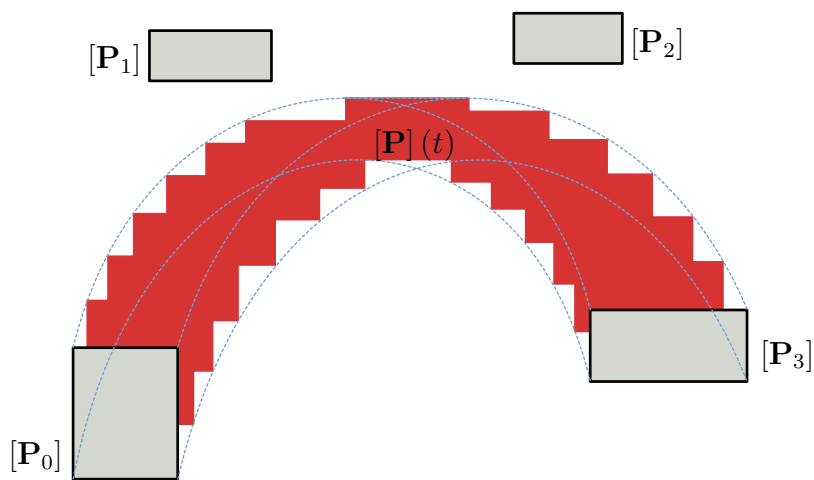
Število operacij, ki jih potrebuje ta način izračuna, je nekoliko odvisen od implementacije metode **EVAL** oziroma od načina izračuna modrih vrednosti v tabeli 3.1. Če za izračun teh vrednosti uporabimo Hornerjev algoritem, potrebujemo skupaj $O(mn)$ operacij.

3.6 Izris intervalskih Bézierovih krivulj

Za zaključek poglavja si oglejmo še dva izmed načinov izrisa intervalskih Bézierovih krivulj. V poglavju 2 smo videli, da lahko intervalske polinome opišemo z dvema polinomoma: polinomom spodnje in zgornje meje območja. Pri intervalskih krivuljah pa je izris malo bolj zapleten, zato si ga oglejmo malo podrobneje.

3.6.1 Izris množice pravokotnikov oziroma kvadrov

Geometrijsko si lahko intervalsko Bézierovo krivuljo predstavljamo kot območje, ki ga obišče spreminjajoči se pravokotnik oziroma kvader med potovanjem po centralni Bézierovi krivulji. Če kvadre oziroma pravokotnike izrišemo za različne t z intervala $[0, 1]$, dobimo približek predstavitve intervalske Bézierove krivulje. Na sliki 3.11 je z rdečo barvo predstavljena intervalska Bézierova krivulja, ki jo dobimo s takšnim načinom izrisa.



Slika 3.11: Prikaz intervalske Bézierove krivulje z izrisovanjem pravokotnikov.

3.6.2 Izris ovojnice intervalske Bézierove krivulje

Če si še enkrat ogledamo sliko 3.2, opazimo, da lahko ovojnico intervalske Bézierove krivulje opišemo z zlepci navadnih Bézierovih krivulj in ravnih odsekov. Ravni odseki se pojavijo na mestih, kjer je odvod krivulje enak 0 oziroma kjer je pravokotnik pri potovanju zamenjal smer gibanja v določeni komponenti (na sliki 3.2 imamo raven odsek v točki, kjer se je naraščanje y komponente spremenilo v padanje). Da bi znali izrisati takšno ovojnico, bomo torej morali poiskati ničle odvoda ter izračunati presečišča Bézierovih krivulj, ki se pojavijo nasproti ravnih odsekov.

3.6.3 Iskanje ničel polinoma v Bernsteinovi bazi

Iskanja ničel polinoma se lahko lotimo s pretvorbo polinoma v običajno bazo in uporabo običajnih postopkov za iskanje ničel, mi pa si bomo ogledali bolj učinkovit način, ki je prilagojen za uporabo v CAGD aplikacijah in te pretvorbe ne potrebuje.

Naj bo $y(x) = \sum_{i=0}^n y_i B_i^n(x)$ polinom stopnje n , zapisan v Bernsteinovi bazi, in naj bo $x \in [0, 1]$. Takšen polinom si lahko predstavljamo kot ravninsko Bézierovo krivuljo stopnje n

$$\mathbf{P}(x) = \sum_{i=0}^n \mathbf{P}_i B_i^n(x),$$

kjer je $\mathbf{P}_i = (\frac{i}{n}, y_i)$. Ker Bézierova krivulja leži v konveksni ovojnici svojih kontrolnih točk, si lahko pri iskanju ničel pomagamo z iskanjem presečišč konveksne ovojnice z x osjo.

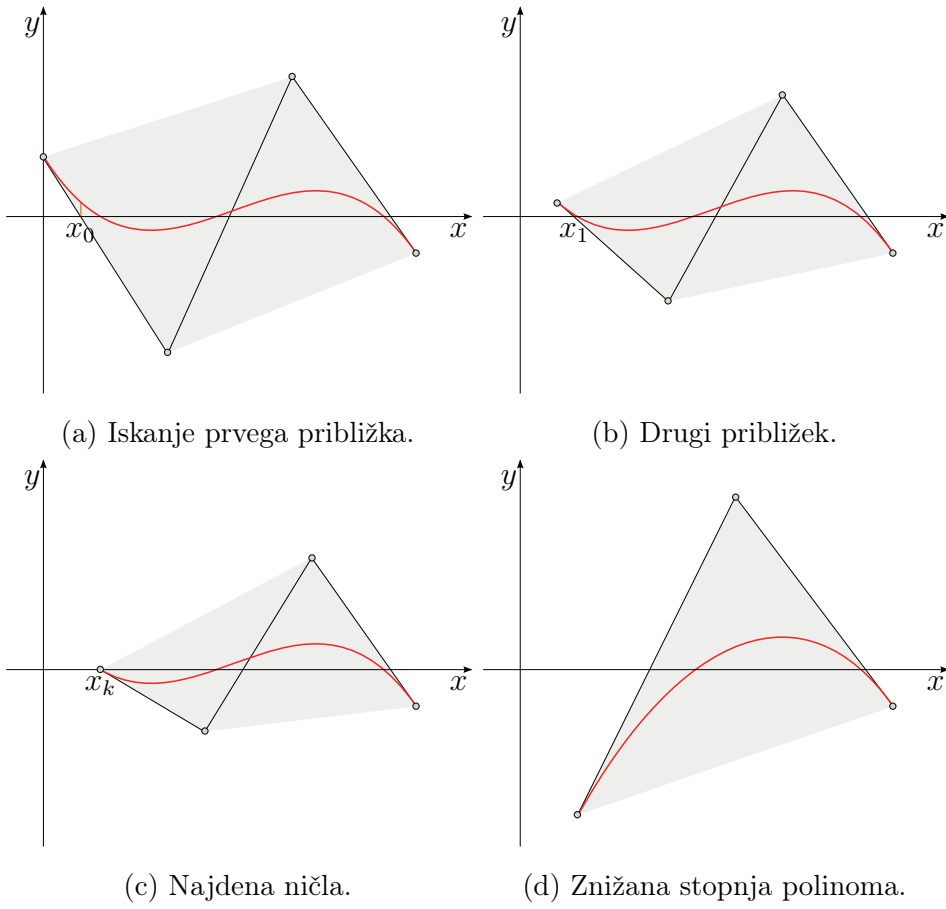
Postopek pričnemo z iskanjem najbolj levega presečišča konveksne ovojnice kontrolnega poligona in x osi, ki ga označimo x_0 (slika 3.12a). Krivuljo s pomočjo de Casteljeujevega postopka v točki x_0 razdelimo na 2 dela in nadaljujemo z desnim delom krivulje (slika 3.12b). S ponavljanjem postopka dobimo približke ničel x_0, x_1, \dots, x_k , ki konvergirajo k skrajno levi ničli polinoma. Ko je $x_{k-1} - x_k < \delta$, x_k označimo za ničlo polinoma (slika 3.12c) in polinomu stopnjo znižamo za ena (slika 3.12d). Zgornje korake ponavljamo, dokler ne najdemo vseh ničel.

Pri nižanju stopnje polinoma želimo ohraniti ničle, ki jih še nismo odkrili, zato mora veljati

$$y(x) = (x - x_k) \bar{y}(x).$$

Koeficiente polinoma $\bar{y}(x)$ lahko izračunamo po formuli

$$\bar{y}_i = \frac{n}{i+1} y_{i+1}, \quad i = 0, 1, \dots, n-1,$$



Slika 3.12: Iskanje ničel polinoma v Bernsteinovi obliki.

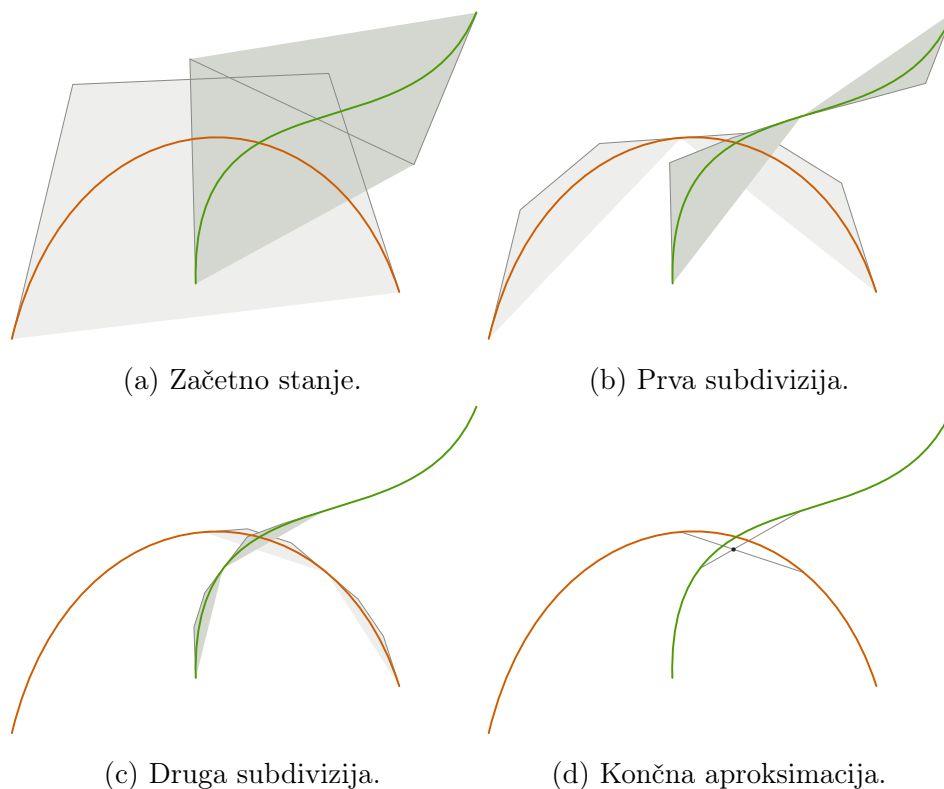
kar sledi iz dejstva, da ima naš polinom prvi koeficient enak 0 in velja

$$\begin{aligned}
 p(t) &= \sum_{i=1}^n p_i B_i^n(t) \\
 &= \sum_{i=1}^n p_i \frac{nt}{i} B_{i-1}^{n-1}(t) \\
 &= t \sum_{i=0}^{n-1} p_{i+1} \frac{n}{i+1} B_i^{n-1}(t),
 \end{aligned}$$

kjer drugi korak sledi iz enačbe (2.4). Pri tem p označuje polinom y , zožen na interval $[x_k, 1]$, in reparametriziran na interval $[0, 1]$.

3.6.4 Iskanje presečišč Bézierovih krivulj

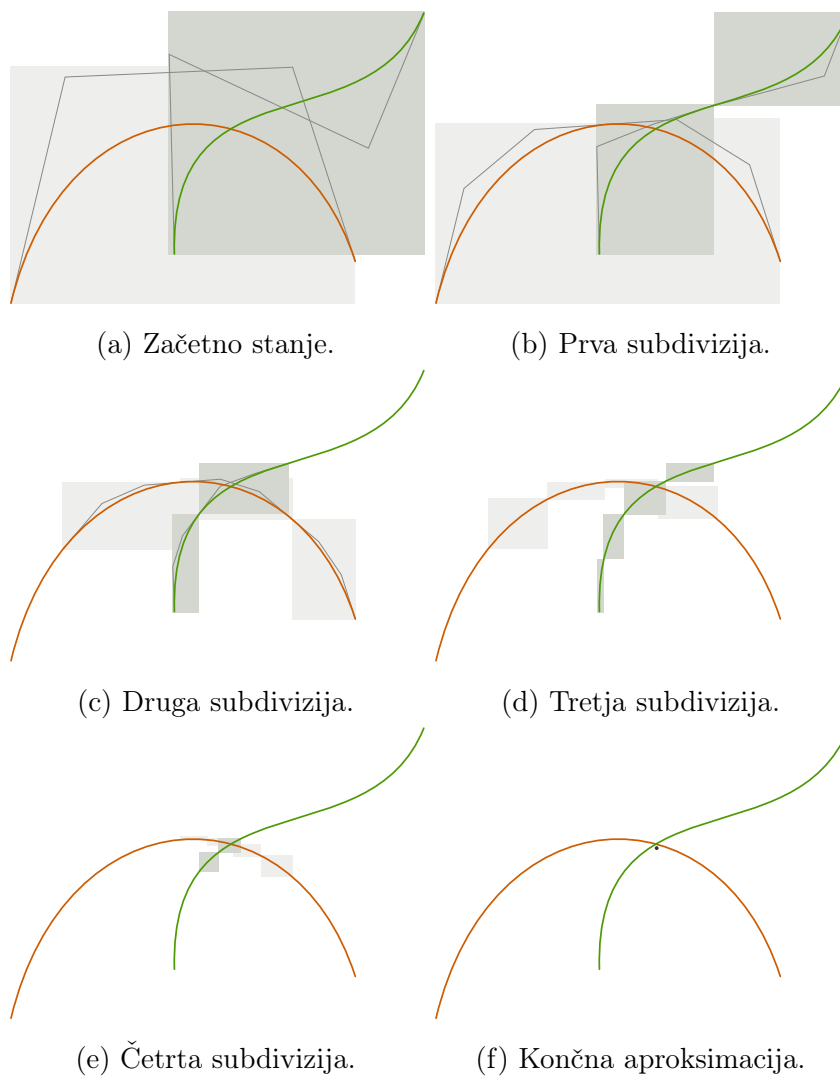
Pri iskanju presečišč dveh krivulj bomo zopet uporabili dejstvo, da Bézierova krivulja leži v konveksni ovojnici svojega kontrolnega poligona. Tako bomo najprej preverili, če se konveksni ovojnici sekata, kar je prikazano na sliki 3.13a. Če se ne, se tudi krivulji ne sekata, zato lahko končamo. V primeru nepraznega preseka pa bomo obe krivulji s de Casteljaujevim algoritmom razpolovili (slika 3.13b) in ponovili postopek na vseh štirih parih dobljenih krivulj. Ko je razdalja med končnima točkama krivulje dovolj majhna, presečišče izračunamo kot presek dveh daljic (slika 3.13d).



Slika 3.13: Iskanje presečišč dveh Bézierovih krivulj.

Določitev preseka dveh konveksnih ovojnici je lahko precej računsko zahtevno, zato bomo v praktični uporabi konveksno ovojnico razširili v pravokotnik oziroma kvader, ki ima stranice vzporedne koordinatnim osem. Iskanje

presekov potem postane trivialno, ker le paroma primerjamo koordinate oglišč. Modificiran algoritem je prikazan na sliki 3.14.



Slika 3.14: Modificirano iskanje presečišč krivulj.

Iz slike lahko opazimo, da ta algoritem potrebuje precej več korakov za dokončanje kot prejšnji, ker se pravokotniki v prvih nekaj korakih večkrat prekrivajo. Zaradi poenostavitve iskanja presekov pa se v praktični uporabi izkaže kot hitrejši.

Poglavje 4

Intervalske Bézierove ploskve

Bézierove ploskve iz tenzorskega produkta bodo osrednja tema tega poglavja. Najprej si bomo ogledali definicijo navadne in intervalske Bézierove ploskve, nato pa bomo ugotovitve iz prejšnjega poglavja uporabili za obravnavo nekaterih lastnosti Bézierovih ploskev ter algoritmov za njihovo manipulacijo. Začnimo pa z definicijo ploskve iz tenzorskega produkta.

4.1 Ploskve iz tenzorskega produkta

Naj bosta f in g funkciji ene spremenljivke. Njun tenzorski produkt h je funkcija dveh spremenljivk, definirana kot

$$h(u, v) = f(u)g(v).$$

Ko parametra u in v pretečeta svoje definicijsko območje, funkcija h v prostoru opiše ploskev, ki jo imenujemo ploskev iz tenzorskega produkta. Intuitivno si lahko ploskev predstavljamo kot sled krivulje, ki se giba po prostoru in spreminja obliko v skladu s funkcijama f in g [4].

Ploskev pa lahko opišemo tudi kot družino krivulj v prostoru. Če v funkciji h fiksiramo parameter $u = c$, dobimo funkcijo

$$h(c, v) = f(c)g(v) = Cg(v),$$

ki določa u -izoparametrično krivuljo. Družina vseh u -izoparametričnih krivulj pa nam določa ploskev. Na podoben način lahko definiramo tudi družino v -izoparametričnih krivulj, ki nam določajo isto ploskev.

4.2 Intervalske Bézierove ploskve

Navadne Bézierove ploskve so zožitev razreda ploskev, ki izhajajo iz tenzorskega produkta. Za krivuljo, ki se giba po prostoru, izberemo Bézierovo krivuljo stopnje m , ki je določena s kontrolnimi točkami $\mathbf{P}_i, i = 0, 1, \dots, m$. Vsaka izmed kontrolnih točk \mathbf{P}_i pa naj se premika po Bézierovi krivulji stopnje n , določeni s kontrolnimi točkami $\mathbf{P}_{i,j}, j = 0, 1, \dots, n$. Takšna ploskev je torej definirana s funkcijo

$$\begin{aligned} \mathbf{P}(u, v) &= \sum_{i=0}^m \mathbf{P}_i(u) B_i^m(v) \\ &= \sum_{i=0}^m \sum_{j=0}^n \mathbf{P}_{i,j} B_j^n(u) B_i^m(v). \end{aligned} \quad (4.1)$$

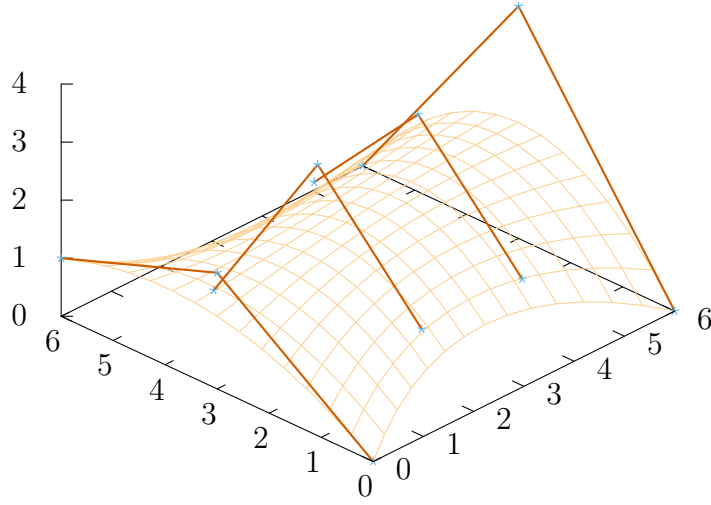
Dobljeno ploskev imenujemo Bézierova ploskev stopnje $m \times n$.

Iz same konstrukcije je jasno razvidno, da so izoparametrične krivulje v tem primeru Bézierove krivulje. Na sliki 4.1 je shematsko prikazana Bézierova ploskev stopnje 3×2 s kontrolnimi točkami.

Intervalske Bézierove ploskve dobimo z enako posplošitvijo, kot smo dobili intervalske Bézierove krivulje. Če v enačbi (4.1) kontrolne točke $\mathbf{P}_{i,j}$ zamenjamo z vektorskimi intervali $[\mathbf{P}_{i,j}]$, dobimo

$$[\mathbf{P}](u, v) = \sum_{i=0}^m \sum_{j=0}^n [\mathbf{P}_{i,j}] B_j^n(u) B_i^m(v).$$

Če v zgornji enačbi vektorske intervale zapišemo v usredinjeni obliki, lahko



Slika 4.1: Primer Bezierove ploskve.

intervalno ploskev izrazimo kot

$$\begin{aligned}
 [\mathbf{P}](u, v) &= \sum_{k=0}^m \sum_{j=0}^n [\mathbf{P}_{k,j}] B_j^n(u) B_k^m(v) \\
 &= \sum_{k=0}^m \sum_{j=0}^n (\bar{\mathbf{P}}_{k,j} + \mathbf{e}_{k,j}[i]) B_j^n(u) B_k^m(v) \\
 &= \sum_{k=0}^m \sum_{j=0}^n \bar{\mathbf{P}}_{k,j} B_j^n(u) B_k^m(v) + \sum_{k=0}^m \sum_{j=0}^n \mathbf{e}_{k,j}[i] B_j^n(u) B_k^m(v) \\
 &= \bar{\mathbf{P}}(u, v) + \mathbf{e}(u, v)[i],
 \end{aligned}$$

kar je analogno rezultatu, ki smo ga dobili pri intervalskih krivuljah. Intervalno Bézierovo ploskev si tako lahko predstavljamo kot navadno Bézierovo ploskev, ki je zadebeljena za $\mathbf{e}(u, v)$.

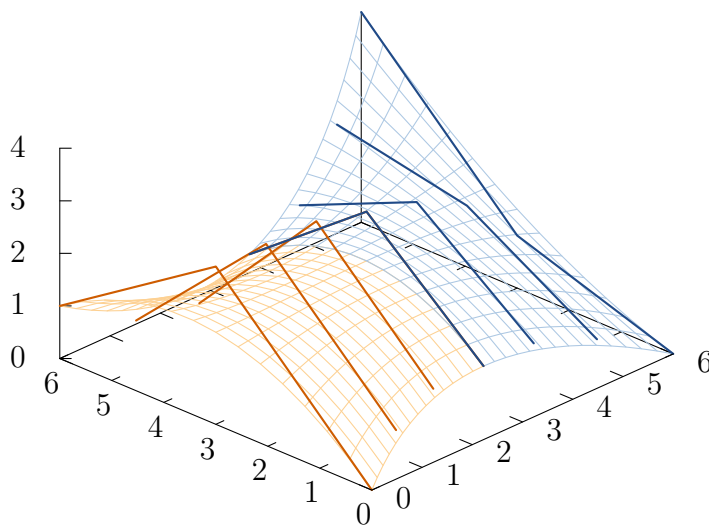
4.3 Subdivizija

Subdivizijo ploskve lahko opravimo na podoben način kot subdivizijo krivulje, če de Casteljaujev postopek uporabimo na vsaki izmed kontrolnih Bézierovih krivulj bodisi v smeri parametra u bodisi v smeri v .

Imejmo Bézierovo ploskev stopnje $m \times n$, ki je določena s kontrolnimi točkami $\mathbf{P}_{i,j}$, $i = 0, 1, \dots, m$, $j = 0, 1, \dots, n$. Če kontrolne točke ploskve zapišemo v matriko

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}_{0,0} & \mathbf{P}_{0,1} & \mathbf{P}_{0,2} & \cdots & \mathbf{P}_{0,n} \\ \mathbf{P}_{1,0} & \mathbf{P}_{1,1} & \mathbf{P}_{1,2} & \cdots & \mathbf{P}_{1,n} \\ \mathbf{P}_{2,0} & \mathbf{P}_{2,1} & \mathbf{P}_{2,2} & \cdots & \mathbf{P}_{2,n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{P}_{m,0} & \mathbf{P}_{m,1} & \mathbf{P}_{m,2} & \cdots & \mathbf{P}_{m,n} \end{bmatrix}, \quad (4.2)$$

potem subdivizijo ploskve v smeri parametra u opravimo s subdivizijo $n + 1$ Bézierovih krivulj, ki jih določajo stolpci matrike \mathbf{P} . Podobno moramo za delitev v smeri parametra v opraviti $m + 1$ subdivizij krivulj, ki jih določajo vrstice matrike \mathbf{P} . Na sliki 4.2 je shematično prikazana subdivizija ploskve na sliki 4.1 pri vrednosti parametra $v = 0.5$.

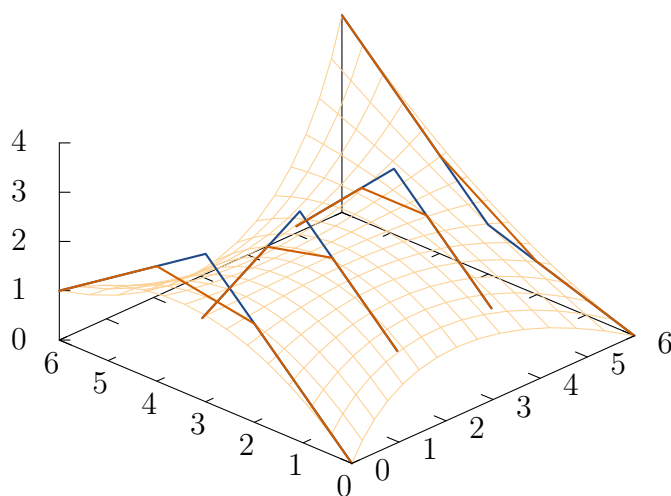


Slika 4.2: Grafični prikaz subdivizije Bézierove ploskve.

Subdivizija intervalske Bézierove ploskve je zopet rezultat trivialne posplošitve, ker subdivizijo krivulj, določenih s stolpci in vrsticami matrike \mathbf{P} zamenjamo s subdivizijami intervalskih krivulj.

4.4 Višanje stopnje

Višanja stopnje Bézierove krivulje se lotimo na enak način kot subdivizije. Če želimo opraviti višanje stopnje v smeri parametra u , potem izvedemo višanje stopnje na Bézierovih krivuljah, ki jih določajo stolpci matrike \mathbf{P} , definirane v (4.2). Podobno velja za višanje stopnje v smeri parametra v . Na sliki 4.3 je prikazano višanje stopnje krivulje s slike 4.1 v smeri parametra u .



Slika 4.3: Grafični prikaz višanja stopnje Bézierove ploskve.

Posplošitev na intervalske krivulje se opravi na enak način kot posplošitev subdivizije.

4.5 Izris

V opisu intervalskih Bézierovih ploskev smo videli, da lahko ploskev opišemo kot družino krivulj. Na osnovi tega dejstva bomo intervalsko Bézierovo ploskev izrisali kot množico ali u - ali v -izoparametričnih intervalskih Bézierovih krivulj. S takšnim načinom izrisa bomo prihranili pri količini programske kode, ki jo moramo spisati.¹

¹Upamo, da bomo s tem pri implementaciji naredili tudi kakšno napako manj.

Poglavje 5

Implementacija algoritmov in rezultati

V tem poglavju bomo na kratko opisali nekatere podrobnosti programske implementacije v prejšnjih poglavjih opisanih metod ter način izrisa rezultatov, nato pa se bomo posvetili prikazom izbranih primerov.

5.1 Izračun točk

Naš glavni cilj, ki smo ga želeli doseči s programsko implementacijo algoritmov, je bila priprava podatkov, ki jih potrebujemo za izris intervalskih krivulj in ploskev. V ta namen smo v programskem jeziku Octave oziroma MATLAB spisali nekaj deset funkcij, ki so priložene k delu.

Implementirane funkcije se v večjem delu ne razlikujejo od postopkov, ki smo jih opisali v prejšnjih poglavjih. Dve izjemi, ki smo jih naredili, pa sta odprava nekaterih zank in zmanjšanje prostorskih zahtev izračunov. Odpravo zank smo dosegli z uporabo vgrajenih programskih konstruktov, ki nam omogočajo množenje komponent vektorjev oziroma matrik z enim samim klicem, pri čemer upamo, da bo tolmač pri izvajanju programa uporabil vektorske ukaze, ki jih premorejo sodobni procesorji. Zmanjševanja prostorskih zahtev pa smo se poslužili pri implementaciji iterativnih operacij, ki za izvedbo na-

slednje ponovitve potrebujejo le zadnji rezultat. Na ta način smo matrične strukture, ki smo jih opisali v algoritmu, skrčili na en sam stolpec oziroma vektor vrednosti.

5.2 Prikaz rezultatov

Rezultate smo grafično upodobili s pomočjo programa Gnuplot. Za vsako vrsto prikaza (2D krivulje, 3D krivulje, ploskve) smo pripravili kratek program, ki vodi izris objekta. Na ta način smo ločili izračun podatkov od njegovega prikaza, ki nam omogoča večjo prilagodljivost. Tako za prikaz objekta iz različnih perspektiv ni potrebno spreminjati glavnega programa, ampak le spremenimo ali dodamo nov Gnuplot program, ki izrisuje na osnovi že obstoječih podatkov.

Sedaj pa si oglejmo nekaj grafičnih prikazov intervalskih krivulj in ploskev, ki so glavni razlog za nastanek tega dela. Začeli bomo s prikazi dvodimenzionalnih intervalskih Bézierovih krivulj, nadaljevali s krivuljami v prostoru, končali pa bomo s prikazom intervalske Bézierove ploskve.

5.3 Ravninske krivulje

Izris krivulje Na sliki 5.1 je prikazana intervalska Bézierova krivulja stopnje 5, podana s kontrolnimi intervali (zapisanimi v usredinjeni obliki)

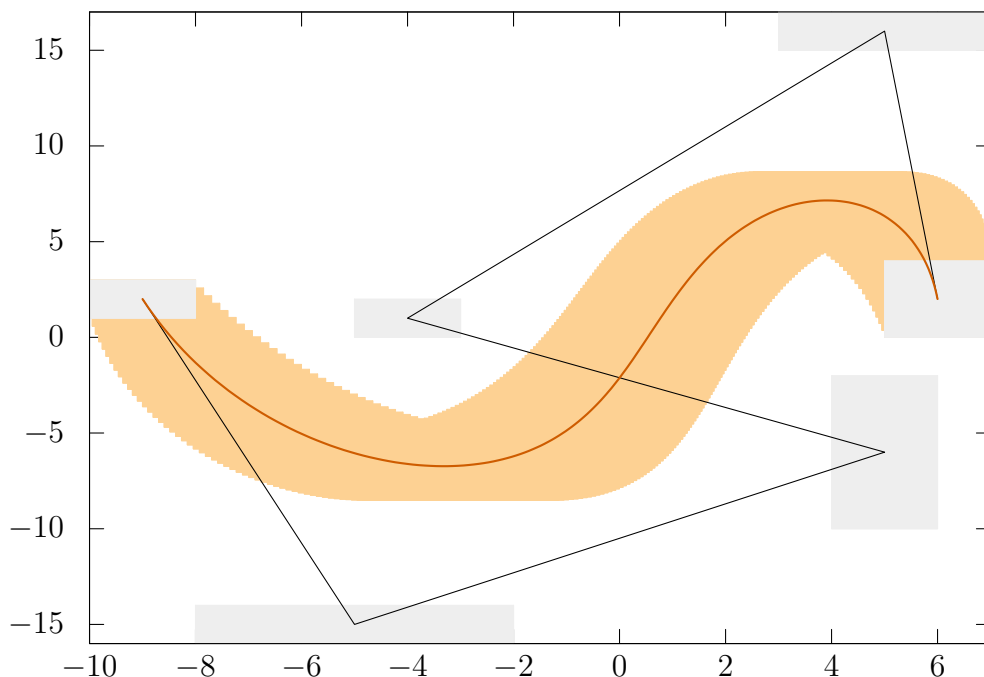
$$\begin{aligned} [\mathbf{P}_0] &= (-9, 2) + (1, 1)[i], & [\mathbf{P}_1] &= (-5, -15) + (3, 1)[i], \\ [\mathbf{P}_2] &= (5, -6) + (1, 4)[i], & [\mathbf{P}_3] &= (-4, 1) + (1, 1)[i], \\ [\mathbf{P}_4] &= (5, 16) + (2, 1)[i], & [\mathbf{P}_5] &= (6, 2) + (1, 2)[i] \end{aligned}$$

in enačbo

$$[\mathbf{P}](t) = \sum_{k=0}^5 [\mathbf{P}_k] B_k^5(t). \quad (5.1)$$

Rumeno označeno območje je intervalska krivulja, sivi pravokotniki predstavljajo kontrolne intervale krivulje, temnejša krivulja, ki poteka v sredini

rumenega območja, pa je navadna Bézierova krivulja, določena s središči kontrolnih intervalov.



Slika 5.1: Krivulja, določena z enačbo (5.1).

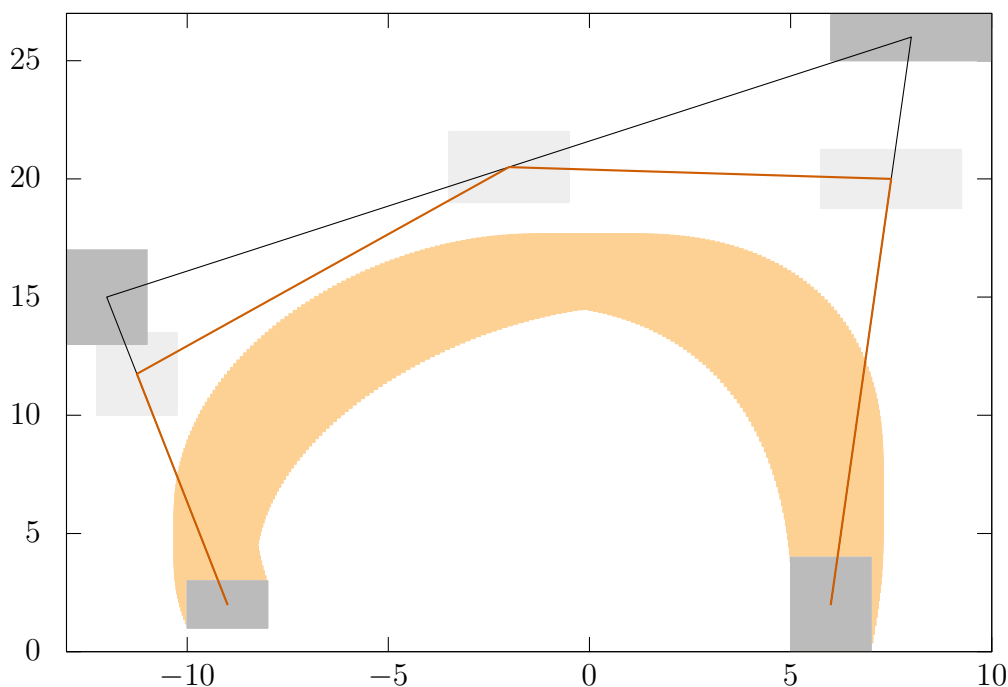
Višanje stopnje Sledi prikaz višanja stopnje intervalske krivulje. Na sliki 5.2 je prikazana krivulja stopnje 3 s kontrolnimi intervali

$$\begin{aligned} [\mathbf{R}_0] &= (-9, 2) + (1, 1)[i], & [\mathbf{R}_1] &= (-12, 15) + (1, 2)[i], \\ [\mathbf{R}_2] &= (8, 26) + (2, 1)[i], & [\mathbf{R}_3] &= (6, 2) + (1, 2)[i] \end{aligned}$$

in enačbo

$$[\mathbf{R}](t) = \sum_{k=0}^3 [\mathbf{R}_k] B_k^3(t). \quad (5.2)$$

Podobno kot prej rumeni del prikazuje intervalsko krivuljo, temnejši pravokotniki predstavljajo kontrolne intervale osnovne krivulje, svetlejši pravokotniki pa so kontrolni intervale krivulje, ki smo jo dobili z zvišanjem stopnje osnovne krivulje.



Slika 5.2: Višanje stopnje krivulje, določene z enačbo (5.2).

Subdivizija Nadaljujemo s prikazom subdivizije intervalske krivulje. Slika 5.3 prikazuje krivuljo s kontrolnimi intervali

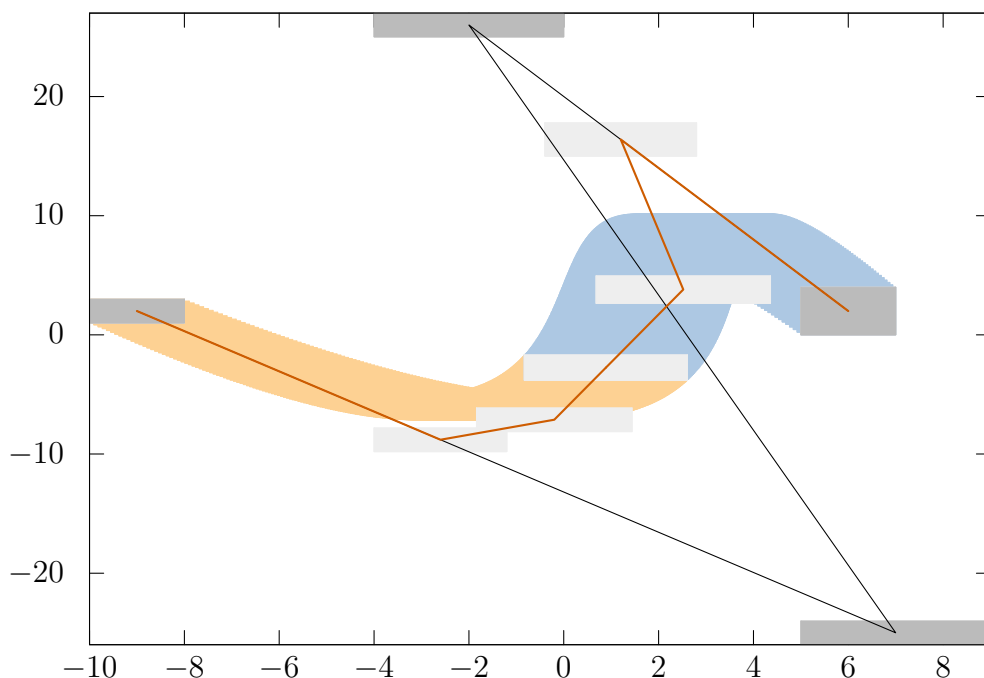
$$\begin{aligned} [\mathbf{Q}_0] &= (-9, 2) + (1, 1)[i], & [\mathbf{Q}_1] &= (7, -25) + (2, 1)[i], \\ [\mathbf{Q}_2] &= (-2, 26) + (2, 1)[i], & [\mathbf{Q}_3] &= (6, 2) + (1, 2)[i] \end{aligned}$$

in enačbo

$$[\mathbf{Q}](t) = \sum_{k=0}^3 [\mathbf{Q}_k] B_k^3(t), \quad (5.3)$$

na kateri smo opravili subdivizijo pri vrednosti parametra $t = 4/10$. Z rumeno in modro barvo sta označeni krivulji, ki ju dobimo s postopkom subdivizije, temno sivi pravokotniki so kontrolni intervali prvotne krivulje, svetlo sivi pravokotniki pa predstavljajo kontrolne intervale novih krivulj.

Obris Naslednja slika pa prikazuje alternativen način izrisa intervalske Bézierove krivulje, in sicer s pomočjo obrisa. Slika 5.4 prikazuje krivuljo s kon-



Slika 5.3: Subdivizija krivulje, določene z enačbo (5.3).

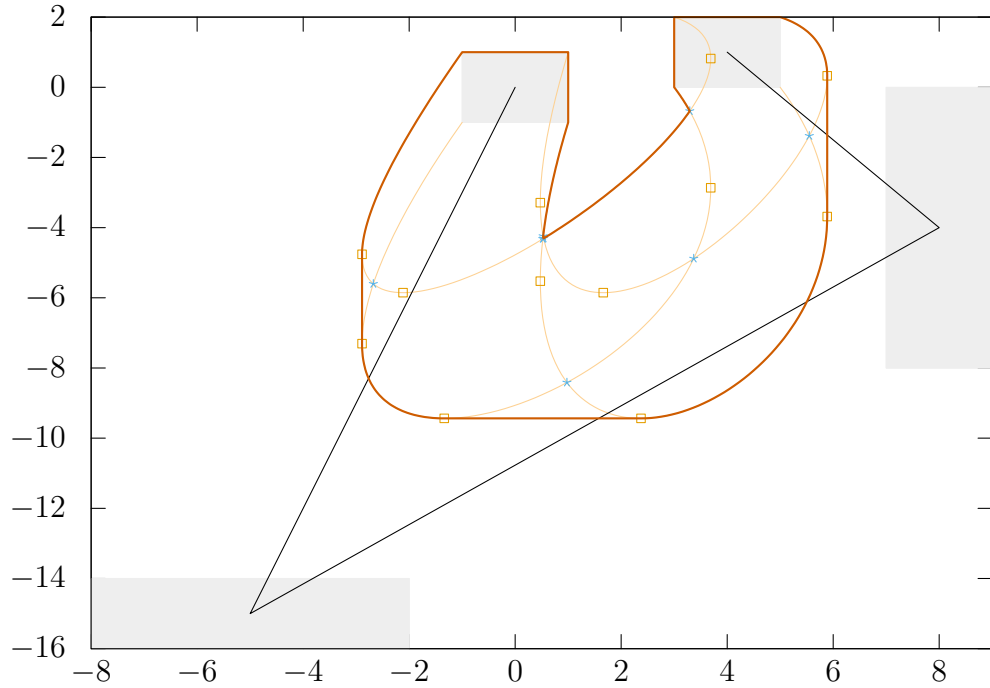
trojnimi intervali

$$\begin{aligned} [\mathbf{S}_0] &= (-0, 0) + (1, 1)[i], & [\mathbf{S}_1] &= (-5, -15) + (3, 1)[i], \\ [\mathbf{S}_2] &= (8, -4) + (1, 4)[i], & [\mathbf{S}_3] &= (4, 1) + (1, 1)[i] \end{aligned}$$

in enačbo

$$[\mathbf{S}](t) = \sum_{k=0}^3 [\mathbf{S}_k] B_k^3(t). \quad (5.4)$$

Postopek izračuna obrisa intervalske krivulje je naslednji. Najprej izračunamo Bézierove krivulje, ki jih opišejo istoležni vogali kontrolnih intervalov (na sliki prikazane z rumeno barvo). Nato izračunamo presečišča teh krivulj (zvezdice) ter vse spremembe smeri gibanja (kvadratki). Po določitvi dveh začetnih krivulj le še orišemo glede na prej izračunane točke.



Slika 5.4: Konstrukcija orisa intervalske krivulje, določene z enačbo (5.4).

5.4 Krivulje v prostoru

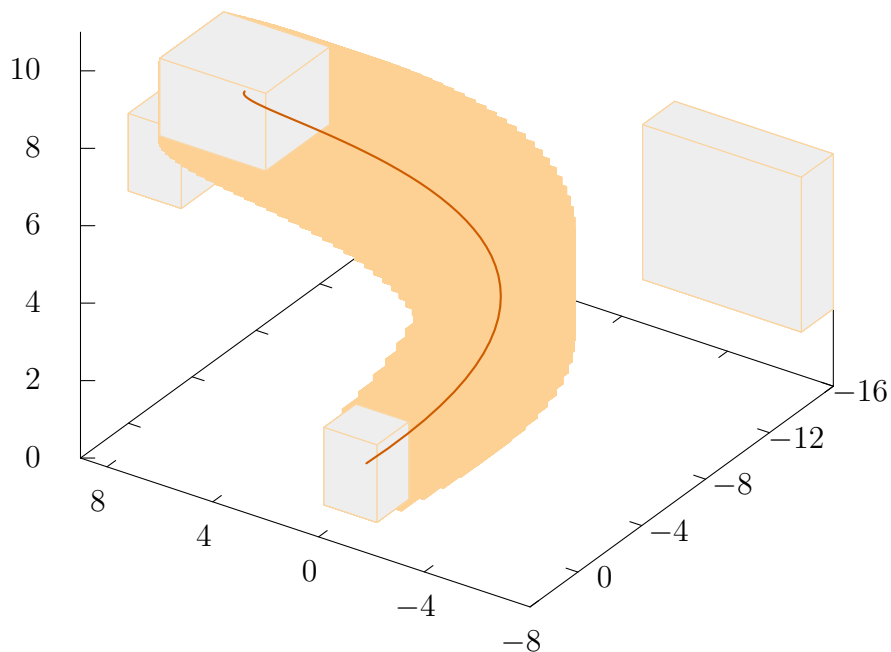
Izris krivulje Začnimo s prikazom preproste intervalske Bézierove krivulje v prostoru. Na sliki 5.5 je prikazana intervalska krivulja s kontrolnimi intervali

$$\begin{aligned} [\mathbf{T}_0] &= (0, 0, 1) + (1, 1, 1)[i], & [\mathbf{T}_1] &= (-5, -15, 4) + (3, 1, 2)[i], \\ [\mathbf{T}_2] &= (8, -4, 7) + (1, 4, 1)[i], & [\mathbf{T}_3] &= (4, 1, 10) + (2, 2, 1)[i] \end{aligned}$$

in enačbo

$$[\mathbf{T}](t) = \sum_{k=0}^3 [\mathbf{T}_k] B_k^3(t). \quad (5.5)$$

Rumeno označeno območje predstavlja intervalsko krivuljo, sivi kvadri predstavljajo kontrolne intervale krivulje, temnejša krivulja v sredini območja pa je Bézierova krivulja, določena s središči kontrolnih intervalov.



Slika 5.5: Intervalska Bézierova krivulja, določena z enačbo (5.5).

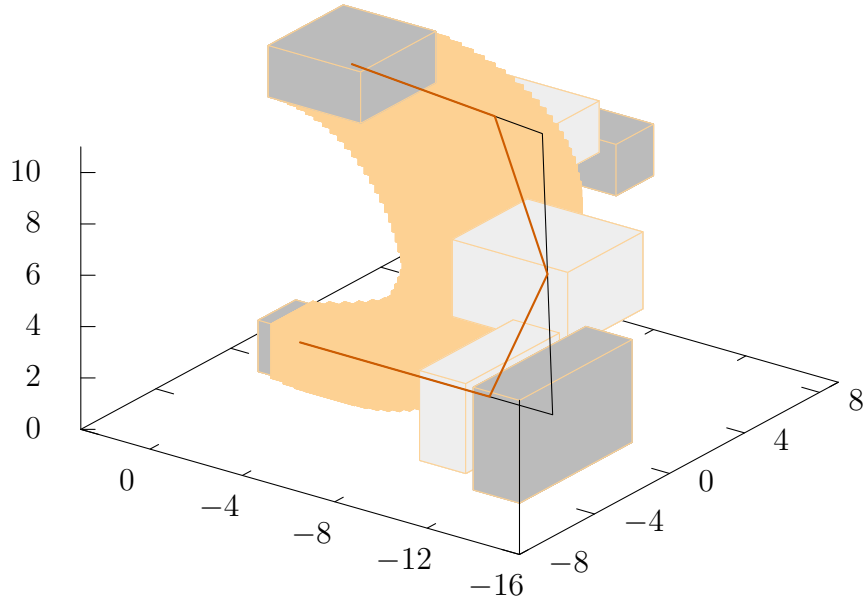
Višanje stopnje Višanje stopnje si bomo ogledali na primeru krivulje, določene s kontrolnimi intervali

$$\begin{aligned} [\mathbf{U}_0] &= (0, 0, 1) + (1, 1, 1)[i], & [\mathbf{U}_1] &= (-5, -15, 4) + (3, 1, 2)[i], \\ [\mathbf{U}_2] &= (8, -4, 7) + (1, 4, 1)[i], & [\mathbf{U}_3] &= (4, 1, 10) + (2, 2, 1)[i] \end{aligned}$$

in enačbo

$$[\mathbf{U}](t) = \sum_{k=0}^3 [\mathbf{U}_k] B_k^3(t). \quad (5.6)$$

Temno sivi kvadri na sliki 5.6 predstavljajo kontrolne intervale osnovne krivulje, svetlo sivi kvadri pa so kontrolni intervali krivulje, ki smo jo dobili z zvišanjem stopnje osnovne krivulje na 4.



Slika 5.6: Višanje stopnje intervalske krivulje, določene z enačbo (5.6).

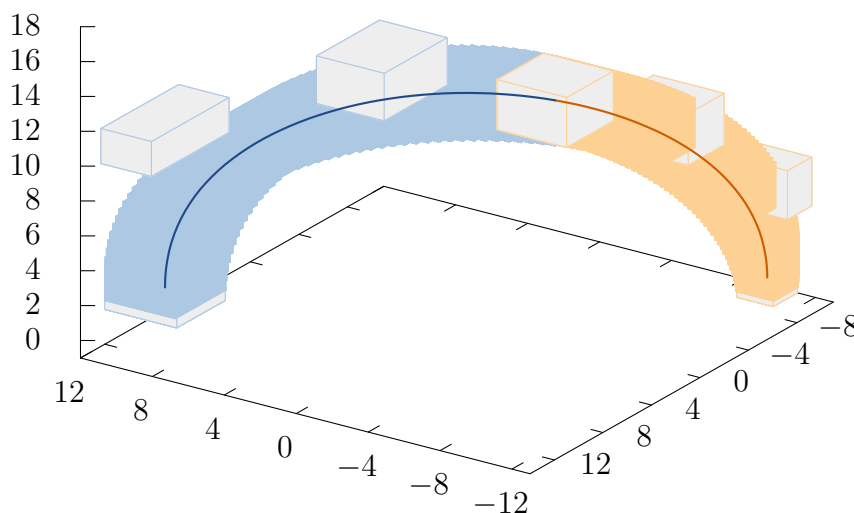
Subdivizija Na sliki 5.7 je prikazana subdivizija krivulje, določene s kontrolnimi intervali

$$\begin{aligned} [\mathbf{V}_0] &= (-10, -10, 0) + (1, 1, 1)[i], & [\mathbf{V}_1] &= (-10, -10, 14) + (3, 1, 2)[i], \\ [\mathbf{V}_2] &= (10, 10, 17) + (1, 4, 1)[i], & [\mathbf{V}_3] &= (10, 10, 2) + (2, 2, 1)[i] \end{aligned}$$

in enačbo

$$[\mathbf{V}](t) = \sum_{k=0}^3 [\mathbf{V}_k] B_k^3(t). \quad (5.7)$$

Z modro in rumeno barvo označeni območji predstavljata intervalski krivulji, ki ju dobimo kot rezultat subdivizije osnovne krivulje pri vrednosti parametra $t = 4/10$. Sivi kvadri predstavljajo kontrolne intervale krivulj, temnejši krivulji v sredini intervalskih krivulj pa sta Bézierovi krivulji, določeni s središči kontrolnih intervalov.



Slika 5.7: Subdivizija krivulje, določene z enačbo (5.7).

5.5 Ploskve

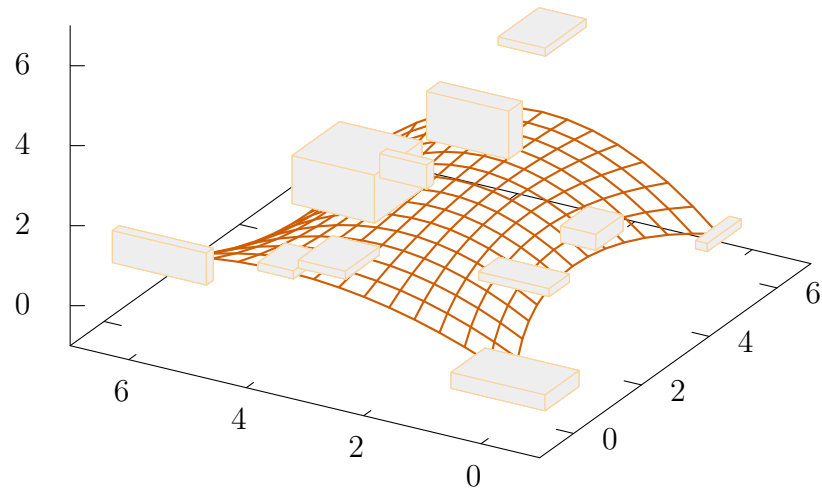
Za konec pa si še oglejmo izris intervalske Bézierove ploskve. Kontrolni intervali naše ploskve naj bodo enaki

$$\begin{aligned}
[\mathbf{Z}_{0,0}] &= (0, 0, 0) + (0.5, 0.8, 0.2)[i], & [\mathbf{Z}_{0,1}] &= (2, 1, 1) + (0.3, 0.6, 0.1)[i], \\
[\mathbf{Z}_{0,2}] &= (4, 1, 1) + (0.4, 0.3, 0.2)[i], & [\mathbf{Z}_{0,3}] &= (6, 0, 0) + (0.5, 0.1, 0.1)[i], \\
[\mathbf{Z}_{1,0}] &= (0, 3, 2) + (0.5, 0.4, 0.1)[i], & [\mathbf{Z}_{1,1}] &= (2, 3, 3) + (0.1, 0.4, 0.3)[i], \\
[\mathbf{Z}_{1,2}] &= (4, 3, 3) + (0.2, 0.7, 0.6)[i], & [\mathbf{Z}_{1,3}] &= (6, 3, 4) + (0.6, 0.4, 0.1)[i], \\
[\mathbf{Z}_{2,0}] &= (0, 6, 1) + (0.1, 0.8, 0.4)[i], & [\mathbf{Z}_{2,1}] &= (2, 5, 0) + (0.4, 0.3, 0.1)[i], \\
[\mathbf{Z}_{2,2}] &= (4, 5, 1) + (0.7, 0.7, 0.6)[i], & [\mathbf{Z}_{2,3}] &= (6, 6, 0) + (0.1, 0.4, 0.4)[i],
\end{aligned}$$

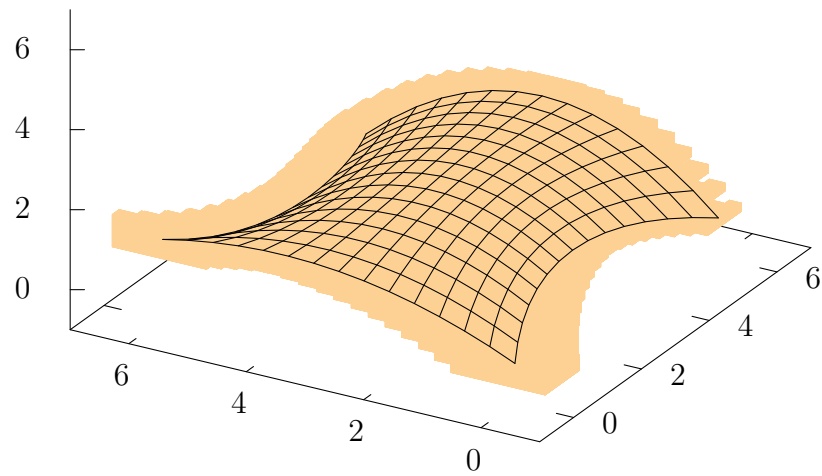
enačba ploskve pa

$$\mathbf{Z}(u, v) = \sum_{j=0}^2 \sum_{k=0}^3 \mathbf{Z}_{j,k} B_k^n(u) B_j^m(v). \quad (5.8)$$

Najprej si na sliki 5.8 oglejmo prikaz kontrolnih intervalov (sivi kvadri) ter ploskev, ki jo določajo središča kontrolnih intervalov (mreža v oranžni barvi). Intervalna ploskev, ki jo določa enačba (5.8), pa je prikazana na sliki 5.9.



Slika 5.8: Kontrolni intervali ploskve, določene z enačbo (5.8).



Slika 5.9: Intervalska Bézierova ploskev, določena z enačbo (5.8).

Poglavje 6

Zaključek

V delu smo si ogledali intervalske Bézierove krivulje in ploskve, področje geometrijskega oblikovanja s pomočjo računalnika, ki je večino časa skrito očem, a ima ključno vlogo pri zagotavljanju kvalitete in ustreznosti končnega izdelka. Opravili smo kratek sprehod skozi teoretične osnove Bernsteinovih baznih polinomov ter intervalske aritmetike, ki nam je priskrbel potrebne osnovne gradnike, s pomočjo katerih smo v nadaljevanju nadgradili Bézierove krivulje in ploskve v njihove intervalske izpeljanke.

Intervalske Bézierove krivulje in ploskve smo opisovali vzporedno z navadnimi Bézierovimi krivuljami in ploskvami, s čemer smo želeli gradivo narediti dostopno začetnikom na področju z računalnikom podprtega geometrijskega oblikovanja, obenem pa poznavalcem navadnih krivulj oziroma ploskev omogočiti lažji prenos obstoječega znanja na intervalske različice. Z grafičnimi prikazi postopkov in končnih rezultatov pa smo želeli bralcu ponuditi dodaten način potrditve razumevanja teorije.

Že obravnavano tematiko tega dela bi lahko v nadaljevanju razširili na mnoga druga področja CAGD. Naravno nadaljevanje bi bila razširitev teorije na zlepke Bézierovih krivulj ali B-zlepke, medtem ko bi analiza uporabe intervalskih Bézierovih krivulj in ploskev za aproksimacijo funkcij predstavljala alternativno smer razširitve.

Literatura

- [1] Gerald E Farin. *Curves and surfaces for CAGD: a practical guide*. Morgan Kaufmann, 2002.
- [2] Gerald E Farin, Josef Hoschek, and Myung-Soo Kim. *Handbook of computer aided geometric design*. Elsevier, 2002.
- [3] O Ismail. Degree elevation of interval bezier curves using legendre-bernstein basis transformations. *International Journal of Video & Image Processing and Network Security (IJVIPNS)*, 10(6):6–9, 2010.
- [4] G Jaklič. Krivulje in ploskve v računalniško podprtem geometrijskem oblikovanju. Lecture notes, 2011.
- [5] Ramon E Moore and RE Moore. *Methods and applications of interval analysis*, volume 2. SIAM, 1979.
- [6] SP Mudur and PA Koparkar. Interval methods for processing geometric objects. *Computer Graphics and Applications, IEEE*, 4(2):7–17, 1984.
- [7] Thomas W Sederberg. Computer aided geometric design. Lecture notes, 2012.
- [8] Thomas W Sederberg and Rida T Farouki. Approximation by interval bézier curves. *IEEE Computer Graphics and Applications*, 12(5):87–95, 1992.